LOW BIT-RATE VIDEO CODING USING WATERSHED SEGMENTATION AND CONTROL POINT TRACKING

by

B Prabhakar

to the

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY
KANPUR

Febrauary, 2000

LOW BIT-RATE VIDEO CODING USING WATERSHED SEGMENTATION AND CONTROL POINT TRACKING

A Thesis Submitted

in Partial Fulfillment of the Requirements

for the degree of

Master of Technology

by

B Prabhakar

to the

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY
KANPUR

Febrauary, 2000

CERTIFICATE

This is to certify that the thesis work entitled **Low Bit Rate Video Coding using Watershed Segmentation and Control Point Tracking** by *B Prabhakar* bearing *Roll No* 9810410 has been carried out under my supervision and the same has not been submitted elsewhere for a degree

(*Dr Sumana Gupta*)
Associate Professor
Dept of Electrical Engineeri
Indian Institute of Technolo

Feb,2000

Acknowledgments

I would like this opportunity to express my sincere thanks to Dr Sumana Gupta, my thesis supervisor for her invaluable guidance and encouragement given to me through out my thesis I am grateful to madam for her suggestions and advises not only in academics but also in my personnel problems She always behaved like a mother and free to discuss my problems

I am very thankful to Ramana for spending so many sleepless nights for making this work successful Thanks to swaroop for helping at every stage when ever I faced some software or computer problems

I thank Silicon Automation Systems (SAS) for their financial support during the fourth semester which helped me a lot

I am very thankful to my inmates Ramana, Kovvi, Madhav Krishna, murali, Prasad, Jai Ram, Shaimila, Swaroop, Manoj, Jana, sekhar vasudeva who made my IIT life memorable

Finally I wish to acknowledge the constant encouragement, support and blessings which my family members gave me

Abstract

In this thesis we describe the design of a low bit rate video codec based on an arbitrary shaped region based approach Unlike conventional region based methods the region shape information is not transmitted in the present approach, as it can be synchronously obtained by segmenting the reconstructed picture at both the encoder and decoder respectively A local decoded picture is divided into several segmented regions and moving regions are selected based on the frame difference The algorithm used for spatial segmentation is a multiscale gradient algorithm followed by the watershed transformation which provides accurate segmentation at very low computational cost A novel corner detection and tracking approach is used for robust motion estimation Corner points are used to represent each moving region and estimation of their motion through tracking is used to characterize the motion of each moving region A Least squares method is used to estimate the motion parameters The estimated motion is used to predict the next frame using motion compensated prediction Finally an efficient method for coding prediction error is also proposed The algorithm developed was tested on standard sequences A data rate between 10 20 kbps was obtained at a frame rate of 7 5 frames/sec The PSNR obtained ranges from 38db to 34db for different sequences indicating good quality of reconstructed images

Contents

List of Figures

Chapter 1

INTRODUCTION

1.1 The need for video compression

The digital representation of an image, or a sequence of image, requires a very large number of bits. The goal of image coding is to reduce the number of bits as much as possible, and to reconstruct a faithful duplicate of the original picture.

The efficient digital representation of image and video signals has been the subject of considerable research over the past 20 years. Digital video coding technology has developed into a mature field and products have been developed that are targeted for a wide range of emerging applications, such as video on demand, digital TV/HDTV broadcasting and multimedia image/video data base services. With the increased commercial interest in video communications, the need for international image and video compression standards arose.

To meet this need, the moving Picture Experts Group(MPEG) was formed to develop coding standards. In recent times MPEG-1 and MPEG-2 video coding standards have attracted world wide attention, due to an increasing number of

very large scale integration (VLSI) and software implementations of these standards becoming commercially available MPEG 4, the most recent MPEG standard for transmitting video signals at very low bit rates is motivated by its potential applications for video phones video conferencing multimedia electronic mail remote sensing electronic newspapers interactive multimedia databases multimedia annotation surveillance telemedicine communication aids for deaf people and many others

The main hurdle in the implementation of these applications arises from the difficulty in compressing huge amount of visual information to meet the available bandwidth range In video phone or video conferencing application the background of the scene remains unchanged in consecutive frames and only few moving regions are present So there is lot of redundant information which is removable As a result the video sequence can be represented by a low bit rate stream which can be transmitted over telephone network whose bandwidth is of the order 64Kb/sec We discuss some of the common video standards in the following section

1 2 Existing Video Standards

1 ITU T H 261 Video Coder

The ITU H 261 Video Coding Standard came into existance in early 80 s This video coder primarily aims at achieving bit rates of $p \times 64$Kbps, where p varies from 1 to 30 Motion compensation is employed to predict the images DCT and VLC are also used The main application are Video telephony and Video Conferencing

2 MPEG 1

MPEG 1 was the next in line after H 261 Many storage media and telecommunication channels like LANS etc are well suited for a bit rate of 1 5 Mbps MPEG 1 satisfies these requirements In addition it is also the first standard to jointly implement both video and audio coders It encourages a lot of interactive applications It doesn't specify any standard encoding process Hence the compression algorithm is left to the designer But the image quality should be comparable to that of a VCR and audio quality to that of a CD

3 MPEG 2

MPEG 2 is an extension of the MPEG 1 International standard for digital compression of video and audio signals During 1990, MPEG recognised the need for a second, related standard for coding video for broadcost formates at higher data rates MPEG 2 standard is capable of coding standard definition television at bitrates from about 3 15Mbps and high definition television at 15 30Mbps It supports a lot of interactive applications like Digital storage media, computer graphics, multimedia etc This generic coding standard is designed to provide a wide spectrum of bit rates, resolutions, quality levels and services The advantage of this system is it's compatability and scalability with other systems The four compatabilities are defined as

- A system is called as backward compatible if an existing decoder can decode a signal encoded by the new encoder
- A system is forward compatible if a new decoder can decode a signal encoded by the present encoder
- The system is upward compatible if a higher resolution decoder is able to decode the signal encoded by the present system

- A system is downward compatible if a lower resolution decoder is able to decode the bit stream or part of the bit stream produced by a higher resolution encoder

MPEG 1 /H 261 are forward and upward compatible with MPEG 2 The standard MPEG 2 is backward and downward compatible with MPEG 1 /H 261 As for as audio is concerned MPEG 2 is only forward and backward compatible Scalability means that a part of a bit stream can be decoded This allows decoders with less processing powers to display video at lower resolution/quality The video standard MPEG 2 is scalable

4 MPEG 4

MPEG 4 was started with a view of achieving bit rates lower than 64 kbps to enable the transmission of video and audio through Public Switch Telephone Networks This is how MPEG 4 evolved beginning from H 261 The final draft is just released

1 3 Very Low Bit Rate Video Coding

A New interest has been recently arisen among the image coding research community in the field of Very Low Bit Rate Image sequence coding The motivation of such an interest lies in the developement of new applications such as video phones, interactive multimedia databases, remote sensing and many others The key requirement for these applications is the low capacity for transmission or storage, in order, to use existing communication networks or for mobile communication

For many years wave form based image coding approaches have been the

only approaches utilized in image compression But they presented severe limitations for very low bit rate low bit rate video coding applications In order to overcome the limitations imposed by first generation image coding techniques, second generation image coding was formally introduced in 1985 This second generation approach consist of taking into account the characteristic of the human visual system to define the coding scheme As a result of including the human visual system, second generation can be also seen as an approach of seeing the image composed by different entities called objects This implies that the image or sequence of images have first to be analysed and/or segmented in order to find the entities

Methods used in second generation coding can be mainly devided into two categories

- Segmentation based schemes
- Model based schemes

1 3 1 Segmentaion-based schemes

Among the different coding approaches grouped under the name of second generation coding techniques, there is an increase of interest in segmentation based image coding approaches These coding methods divide the images into a set of connected regions so that every pixel in the image is related to one, and only one, region The set of regions form a partition of the image Each region in the partition recieves a different label Images are described in terms of the partition as well as the information related to the interior of each region(e g texture, motion, etc) This information is neccessary to reconstruct the image in the reciever The reason for the current interest in segmentation based coding approaches is mainly twofold

First they are the basis for the new efficient coding methods Being a second generation technique segmentation based coding approaches try to eliminate the redundant information within and between frames taking into account the special properties of the human visual system In particular the segmentation procedure yield a partition whose regions are homogeneous in some sense Due to this homogeneity, the information of each region can be separately coded in a very efficient manner

Secondly, they open the door to new functionalities in the coding scheme In the framework of video coding, new coding schemes allowing functionalities such as content based multimedia data access etc Such functionalities demand a description of the image sequences in terms of objects which can be grouped into areas of interest by the user

1 3 2 Model-based schemes

In model based image coding the input image is viewed as a 2 D projection of a 3 D real world (scene) The coding is performed by first modeling the 3-D scene, extracting the model parameters at the encoder and finally synthesizing the image at the decoder by using the extracted and quantized parameters If we can reconstruct the three dimensional scene model that leads to 2 D image sequence, and the images are analyzed and synthesized based on this model, then a great reduction in image information can be expected This is the idea of *Model – based coding method* However they can be used for a very limited range of scenes

Though these coders offer the promise of fairly good quality images at low bit rates, the complete system is still under study and the coders designed require considerable computational resources[19] Need for a good model is very important

since the quality of the image reconstruction depends to a large extent on the model Further more the reported systems are limited to a particular model such as the head and shoulder image and a stationary back ground A system using a model of Claire cannot provide good results for the Miss America sequence

1 4 Objective of the thesis

In this thesis our objective was to build a Low bit rate PC BASED VIDEO CODEC for bit rates in the range of 20kbps or less Inspired by different techniques used to achieve low bit rates a new algorithm has been developed and described in this thesis Using the new method reconstructed images of good quality was achieved for a bit rate of 10 20kbps

Basically the method developed is a segmentation-based coding system involving three stages

1 In the first stage *Segmention* is carried out that splits the original data into various homogeneous components such that each component corresponds as much as possible to semantic units Morphological based *Watershed Segmentation Algorithm* has been used for this purpose Mathematical morphology is indeed attractive for this purpose because it is a geometrical approach to signal processing and easily deals with criteria such as shape, size, contrast, connectivity, etc Moreover, Mathematical morphological transformations can be efficiently implemented in both software and hardware This is of prime importance, because the major bottleneck in the segmenation based coding schemes is the complexity, accuracy and computational load of the segmentation stage The *Watershed Algorithm* however can potentially provide anaccurate segmentation at low computational cost[3]

2 The second stage is the *Motion estimation* Although the segmentation based

video coding techniques rely on the concept of arbitrarily shaped regions the motion estimation methods used in H 261 MPEG 1 etc rely mainly on block matching techniques The problems associated with conventional block based approaches are a) Block distortion due to block based processing in prediction and in Transform coding frequently appears at low bit rates because the prediction errors cannot be encoded sufficiently b) Some times appearance of frozen blocks can cause serious distortion

In the present algorithm Low level features are used for motion estimation The low level features such as the corner points of the image are extracted and their position is tracked between frames Feature based approach reduces the vast amount of data present in an image without necessarily eliminating salient information Low level descriptors are preferred for important reasons like generality and graceful degradation

3 In the third stage *coding* is carried out This consists of coding the prediction error and motion information The standard JPEG coding scheme for prediction error coding is not suitable for very low bit rate applications(10 20Kbps) because it takes around 50kbits to represent the error information itself In this thesis we propose a new coding scheme employing standard DCT and quantization for prediction error coding

1 5 Organization Of The Thesis

This thesis is organized as follows

1 **Chapter 2** This chapter gives an overview of the adopted approach to design the low bit rate video codec

2 **Chapter 3** This chapter describes the Multiscale gradient based watershed segmentation algorithm for accurate segmentation The use of Multiscale gradient algorithm in enhancing the blurred edges over noise is discussed An efficient algorithm for the removal of small local minima is explained This is followed by a discussion on implementation of Immersion algorithm to calculate the fast watersheds

3 **Chapter 4** This chapter discusses a new motion estimation algorithm based on the control point tracking approach The selection of moving regions from the segmented regions of the previous frame and 3 D motion estimation of each moving region interms of eight motion parameters is discussed

4 **Chapter 5** This chapter discusses an improved version of standard JPEG method to code the error image

5 **Chapter 6** Results obtained at each stage of the coder are discussed and PSNR of different image sequences are compared for different bit rates

6 **Chapter 7** This chapter concludes the thesis and discusses the scope for future work

Chapter 2

APPROACH ADOPTED IN THE DESIGN OF LOW BIT RATE VIDEO CODEC

The approach adopted, in the design of the low bit rate video (< 20kbps) encoder and decoder, utilizes a hybrid coding scheme of motion compensated prediction and transform coding. The difference between conventional H.261, H.263, MPEG1, MPEG2 standard coding methods and the proposed algorithm is the use of region based methods to obtain arbitrary shaped regions of an image for coding instead of square block structure normally used.

The encoding process is illustrated in Figure 2.1. A local decoded picture is devided into several segmented regions in the segmentation block. Moving regions are selected from the segmented regions based on a frame difference calculated in the changed pixel detector, between a current input frame and a previous decoded frame. The 3-D motion of each moving region is estimated interms of eight motion parameters. Then a predicted picture is constructed by motion-compensation pre-

diction for every moving region Finally the significant prediction error regions are encoded by a modified transform coding algorithm The proposed method to encode the error picture is found to be suitable for low bit rate applications In addition the estimated motion parameters are encoded using uniform quantization and fixed length coding

The decoding process is shown in Figure 2 2 Decoder recieves the serial bits and decodes it to recover the error and motion information The previous decoded picture is segmented using method used in the encoder The predicted picture is constructed by motion compensated prediction based on the decoded motion parameter information Finally, the prediction error is added to the predicted picture to get the reconstructed original frame

This approach is inherently much more simpler than other approaches described in the literature The main advantage is that it is not necessary to send the boundary information of the main regions which in reality takes the largest fraction of the band width in other approaches Using the method described a bit rate as low as 9 2Kbps has been achieved for a SNR of 37db

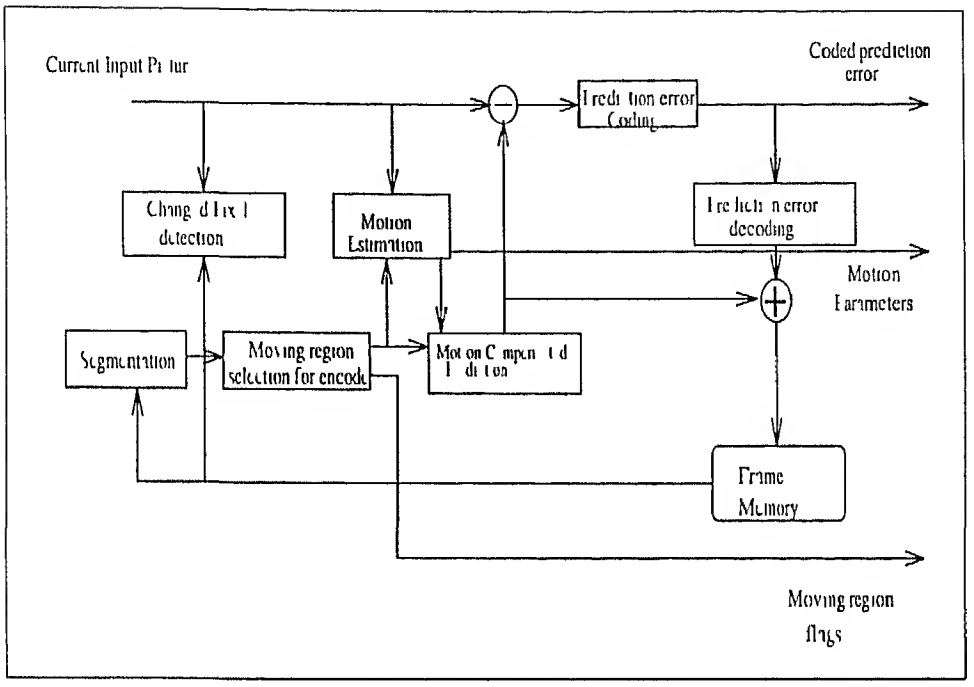


Figure 2.1 Block diagram of Coder

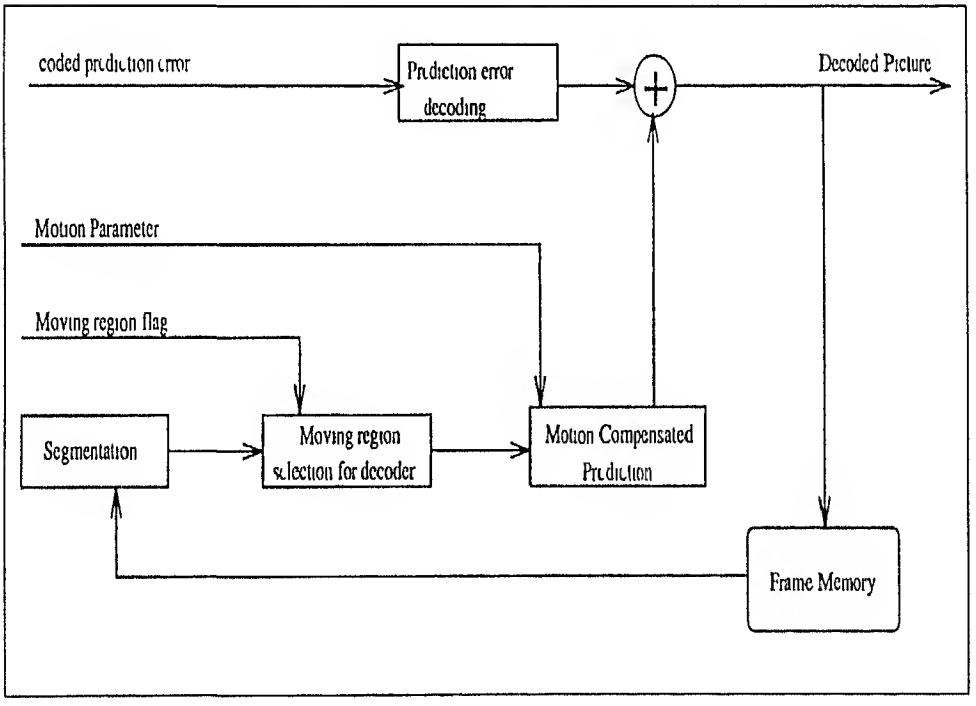


Figure 2.2 Block diagram of Decoder

Chapter 3

SEGMENTATION

The goal of image segmentation is to partition an image into homogeneous regions and locate the contours of the regions as accurately as possible Hence segmentation is as essential step in any Region based method

The segmentation methods in general can be classified into three groups thresholding, edge based and region based segmentation

1 **Thresholding** Thresholding represents the simplest image segmentation method, and is computationally inexpensive and fast Global thresholding i e using a single threshold for the entire image is successful only under special circumstances when gray level variations in an image is small In other cases, segmentation methods using variable thresholds, that is the threshold value is varied over the image as a function of the local image characteristics, can be a better choice some extent Different threshold detection methods exist to determine respective thresholds automatically Threshold based segmentation techniques will fail if the assumptions made in the detection of thresholds are not true

2 **Edge based** Edge based segmentation relies on edges found in an image by edge detection operators These edges mark the image locations of discontinuities in grey level color, texture etc The most common problem that arises in many edge detection technique is that the edges that are detected are false At some times detected edges often have gaps between them at locations where the transition between regions are not abrupt enough So detected edges may not necessarily form a set of closed connected curves that surround connected regions

3 **Region based** In these methods, image segmentation is carried out dividing the image into different regions The different methods are region merging, region splitting and split and merge One of the drawbacks with the region extraction method is that it processes the image in an iterative manner and usually requires large computational time and memory

Recently morphological based watershed algorithm has been developed for image segmentation Mathematical morphology is indeed attractive for the segmentation purpose because it is a geometrical approach to signal processing and easily deals with criteria such as shape, size, contrast, or connectivity that can be considered as segmenation oriented features [4] Morphological approach to image segmentation combines region growing and edge detection techniques It groups the image pixels around the region minima of the image and the boundaries of adjacent groupings are precisely located along the crest lines of the gradient image This is achieved by a transformation called *watershed transformation* Watershed Transformation is a powerful tool for image segmentation which can potentially provide accurate segmentation with very low computational cost as compared segmentation algorithms

For rhe low bit Rate video codecs using segmentation based methods, accurate seg menation of a frame is an important requirement In the proposed video codec design we used the Morphologic al b ised *Water shed Segmentation* algorithm[2]

For image segmentation watershed transformation starts with the gradient of the image to be segmented It views the gradient image as a three dimensional (3 D) surface where the gradient values act as surface heights Intensity edges in the image to be segmented gener ally h ive high gr adient v alues which will appear as *watershed lines* on the 3 D surface while the interior of each region usu ally has a low gr adient value which is consider ed as a *catchment basin* on the 3 D surface The watershed lines partition the gradient image into different catchment basins which correspond to homogenous regions of image to be segmented Watershed transfor mation involves a search for watershed lines in the gradient image Therefore, the performance of a watershed based image segmentation method largely depends on the algorithm used to compute the gradient

Conventional gradient algorithms exhibit a serious weakness for watershed based image segmentation A conventional gradient operator such as the first order partial derivative of Gaussian filter and morphological gradient operators produce too many local minima because of noise and quantization error with in homogenous regions Each minimua of the gradient introduces a catchment basin of the watershed transformation As a result,these gradient operators result in over segmentation ie homogenous regions are partitioned into a large number of regions and proper contours are lost in a multitude of false ones

We choose Multiscale gradient algorithm [3] based on morphological oper ators to reduce the oversegmentation This algorithm efficiently enhances blurred edges such that their gradient values increase above those caused by noise and quan

tization error Then by using an algorithm to eliminate the local minima and giving this multiscale gradient image as input to the watershed transform We got good results

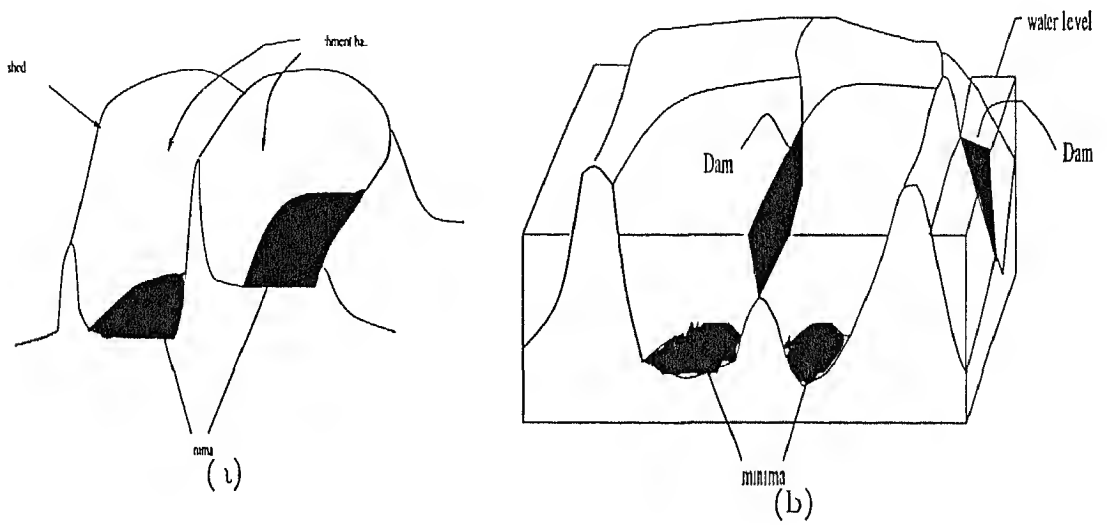


Figure 3 1 (a)Minima,catchment basins and watersheds (b) Building dams at the places where the water coming from two catchment basins would merge

3 1 Morphological tools of Interest

In this section we briefly describe some morphological tools of interest for the total segmentation algorithm

Dilation Let f(x,y) denote input image and b(x,y) the structuring element The grey scale dilation of f by b, denoted by $f \oplus b$, is defined as

$$(f \oplus b)(s,t) = max\{f(s-x,\, t-y) + b(x,y) | (s-x), (t-y) \in D_f, (x,y) \in D_b\}$$

Erosion If f(x,y) is the input image and b(x,y) a structuring element, then Grey scale Erosion of f by b, denoted by $f \ominus b$, is defined as

$$(f \ominus b)(s,t) = min\{f(s+x, t+y) - b(x,y) | (s+x)\ (t+y) \in D_f, (x\ y) \in D_b\}$$

Geodesic dilation A geodesic dilation involves two images a marker image and a mask image Let f denote the marker image and g the mask image such that $f \leq g$ The geodesic dilation of size 1 of the marker image f with respect to the mask image g is denoted by $\delta_g^{(1)}(f)$ and is defined as the point wise minimum between the mask image and the elementary dilation of the marker image It can be expressed as

$$\delta_g^{(1)}(f) = \delta^{(1)}(f) \wedge g$$

Geodesic erosion The geodesic erosion is the dual transformation of the geodesic dilation wih respect to set complementation

$$\varepsilon_g^{(1)}(f) = \varepsilon^{(1)}(f) \vee g$$

where $f \geq g$ and $\varepsilon^{(1)}$ is the elementary erosion Hence, the marker image is first eroded followed by the point wise maximum with the mask image

Reconstruction by erosion The reconstruction by erosion of a mask image g from a marker image f $(f \geq g)$ is defined as the geodesic erosion of f with respect to g until stability is reached It is denoted by $R_g(f)$

$$R_g^*(f) = \varepsilon_g^{(i)}(f)$$

where i is such that $\varepsilon_g^{(i)} = \varepsilon_g^{(i+1)}(f)$

Geodesic Distance Let $A \subset \mathcal{Z}^2$ be a set which is the gray scale image and domain D_A, x, and y are two points of A The geodesic distance $d_A(x,y)$ between two pixels x and y in A is the infimum of the length of the paths which join x and y and are totally included in A

$$d_A(x,y) = inf\{l(P) \;\; P \text{ path between } x \text{ and } y \text{ which is totally included in } A\}$$

Geodesic Influence Zone Suppose the set A contains a set B made of several connected components B_1 B_2 B_k The geodesic influence zone $iz_A(B_i)$ of a connected component B of B in A is the locus of the points of A whose geodesic distance to B_i is smaller than their geodesic distance to any other component of B

$$iz_A(B_i) = \{p \in A, \forall j \in [1,k]/\{i\}, d_A(p,B_i) < d_A(p,B_j)\}$$

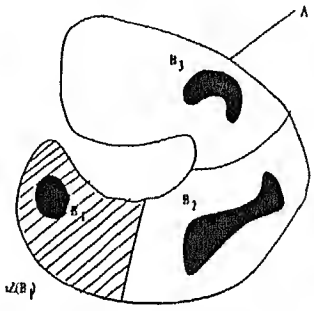


Figure 3 2 Geodesic influence zone of connected component B_1 inside set A

3 2 Multiscale Gradient Algorithm

Ideal step edges do not exist in natural images since every edge is blurred to some extent A blurred edge can be modelled by a ramp and the intensity change between two sides of the edge is referred to as edge height For a ramp edge, the output of a conventional gradient operator is the slope of the edge as shown in figure 3 3 Hence, the ramp edge cannot be separated from noise and quantization error by thresholding, if the slope of the edge is small So a suitable gradient operator for watershed transformation is chosen as the one whose output is equal to the input edge The general morphological gradient operator is given by

$$G(f) = (f \oplus B) - (f \ominus B) \tag{3 1}$$

Where $\oplus$ and $\ominus$ denote dilation and erosion operations respectively and B is called structuring element This gradient operator is referred to as monoscale gradient operator Its performance depends on the size of the structuring element B If B is large the output of this gradient operator for ramp edge is equal to the edge height But large structuring elements result in serious interaction among edges which may lead to gradient maxima not coinciding with the edges However if the structuring element is very small this gradient operator has a high spatial resolution, but produces a low output value for ramp edges In order to exploit the advantages of both large and small structuring elements, a multiscale morphological gradient algorithm is proposed [3]

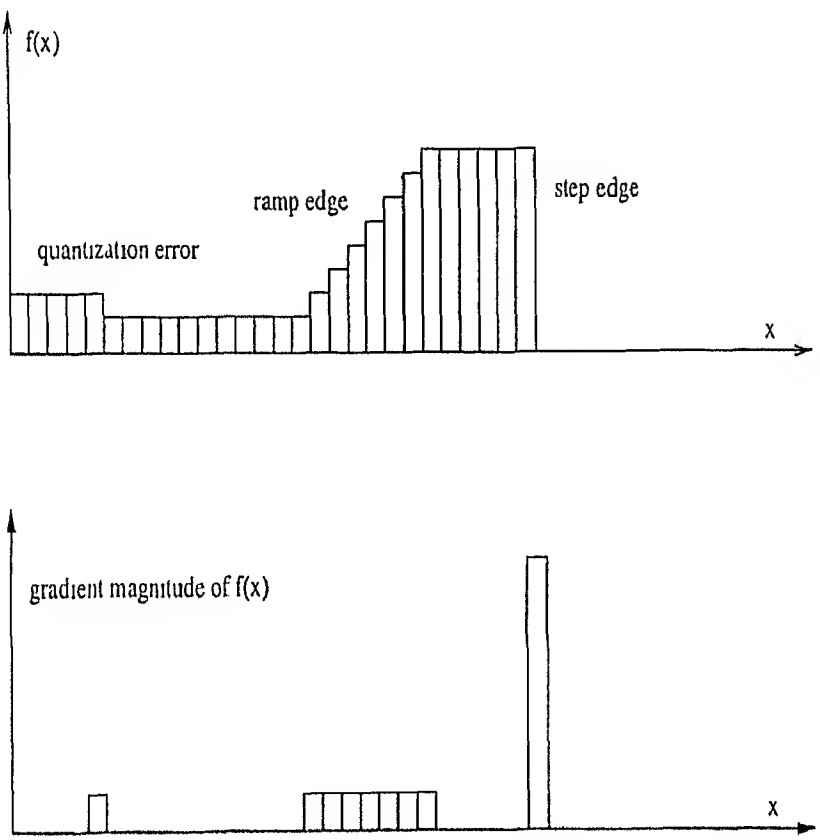


Figure 3 3 Output of conventional gradient operators

Let B for $0 \leq i \leq n$, denote a group of square structuring elements The size of B_i is $(2i + 1) \times (2i + 1)$ pixels Then Morphological multiscale gradient is defined by

$$MC(f) = \frac{1}{n} \sum_{i=1}^{n} [((f \oplus B_i) - (f \ominus B)) \ominus B_{-1}] \quad (3\ 2)$$

For a step edge the operation $((f \oplus B_i) - (f \ominus B_i)) \ominus B_{i-1}$ produces a line of two pixel wide which coincides with the edge and intensity (height) of the line is equal to the edge height Hence the multiscale gradient is equivalent to a monoscale gradient operator in this case, but is more robust to noise due to the averaging operation used in the algorithm For ramp edge of width w and height h, the operation $((f \oplus B_i) - (f \ominus B_i)) \bigcirc B_{i-1}$ produces a line coinciding with the edge The cross section of the line appears as a trapezoid for small i and as a triangle otherwise[3] The width of the bottom side of the trapezoids or triangles is always equal to $w + 2$ pixels with heights always greater than the edge slope h/w [3] Therefore, the multiscale gradient responds effectively to ramp edges without enlarging edges

3 3 Elimination of small local minima

A small local minima is defined as that local minima consisting of a small number of pixels or having a low contrast with its neighbours This kind of local minima in gradient images is generally caused by noise or quantization error, and therefore should be eliminated such local minima are eliminated by dilation with a square structuring element B_s of 2×2 pixels denoted by $(MG(f)) \oplus B_s$ [3], and adding a constant h to the dilated gradient image The local minimas with a contrast lower than h can then be filled up using the reconstruction by erosion of $MG(f)$ from $((MG(f)) \oplus B_s + h)$ Consequently the final gradient image can be expressed as

$\phi^{(rec)}[(MG(f)) \oplus B_s + h \;\; MG(f)]$

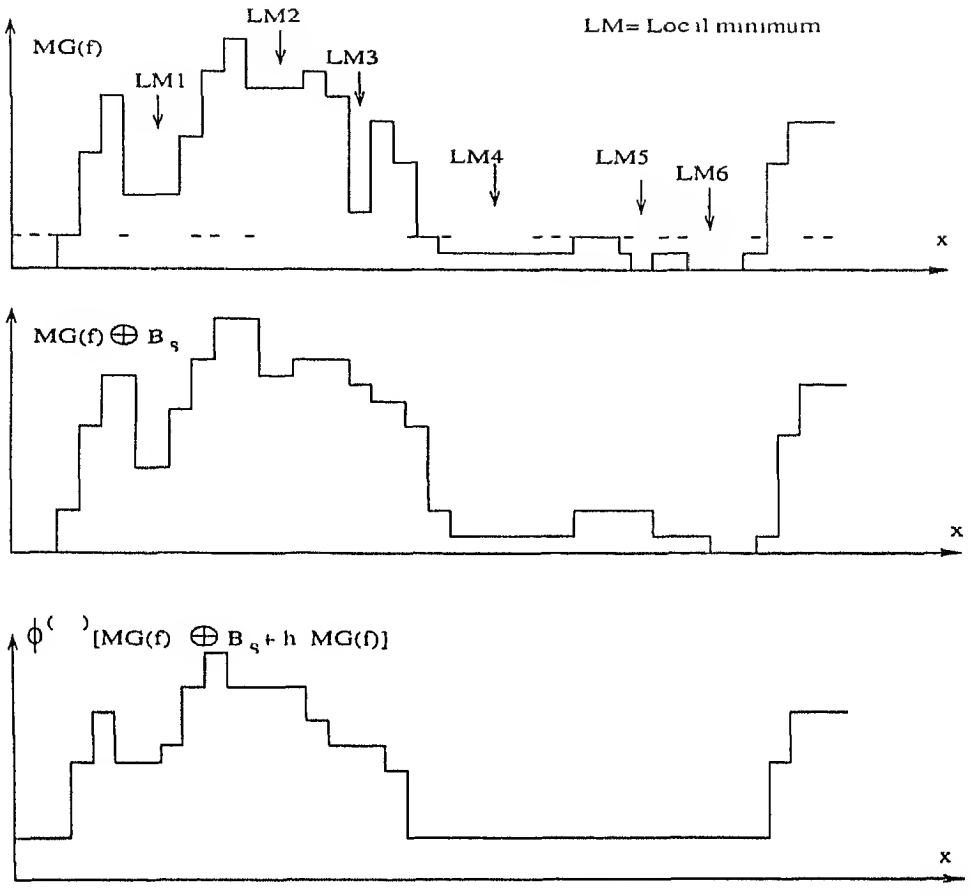


Figure 3 4 Elimination of small local minima

3 4 Watershed Alogorithm

The concept of *watersheds* and *catchment basins* are well known in topography The North American Continental divide is a text book example of a watershed line with catchment basins formed by the Atlantic and Pacific Oceans In the field of Image processing, grayscale images are often considered as topographic reliefs In the topographic representation of a given image the numerical value of each pixel stands for the elevation at this point

Let us consider the topographic representation of a grey level image Let a drop of water fall on such a topographic surface According to the law of gravitation it will flow down along the steepest slope path until it reaches a minimum The entire set of points on the surface whose steepest slope paths reach a given minimum constitutes the *catchment basin* associated with this minimum The *watersheds* are the zones dividing adjacent catchment basins

3 4 1 Some Definition in terms of flooding simulations

The definition of watersheds in terms of water flow is not suitable for well- suited to an algorithmic implementation as there are many cases where the flow direction at a given point is not determinable (e g flat regions etc) For digital functions, there exists no rule to set up the path a drop of water would follow [2] Hence a definition interms of flooding simulations that alleviates these problems is used

Consider again the grey tone image as a topographic surface and assume that holes have been punched in each regional minima of the surface The surface is then slowly immersed into a lake Starting from the minima at the lowest altitude, the water will progressively flood the catchment basins of the image In addition, dams are erected at the places where the waters coming from two different minima would merge At the end of this flooding procedure, each minima is completely surrounded by dams, which delineate its associated catchment basin The resulting dams corresponds to the watersheds They provide us with a partition of the input image into its different catchment basins

To simulate this immersion procedure, we start from the set $T_{h_m}(I)$, the points of which being the first to be reach by the water These points constitute the

starting set of our recursion We thus put

$$X_h = T_{I_m}(I) \tag{3.3}$$

X_h is made of the points of I which belong to the minima of lowest altitude Let us now consider the threshold of I at level $h_{min}+1$ i.e., $T_{h_{n+1}}(I)$ Obviously $X_{h_m} \subseteq T_{h_{+1}}(I)$ Now Y being one of the connected components of $T_{h_{+1}}(I)$, there are three possible inclusion relations between Y and $Y \cap X_{h_{m_n}}$

1 $Y \cap X_h = \emptyset$ In this case, Y is obviously a new minimum of I

2 $Y \cap X_h \neq \emptyset$ **and is connected** In this case, Y corresponds exactly to the pixels belonging to the catchment basin associated with the minimum $Y \cap X_{h_m}$ and having gray level lower or equal to $h_{min}+1$

$$Y = C_{h_{m\ +1}}(Y \cap X_{h_{m\ n}}) \tag{3.4}$$

3 $Y \cap X_{h_m} \neq \emptyset$ **and is not connected** we notice that Y contains different minima of I Denote by $Z_1, Z_2, Z_3, \ldots, Z_k$ these minima, and let Z_i be one of them At this point, the best possible choice for $C_{h_{m\ n+1}}(Z_i)$ is given by the geodesic influence zone of Z_i inside Y

$$C_{h_{m\ +1}}(Z_i) = iz_Y(Z_i) \tag{3.5}$$

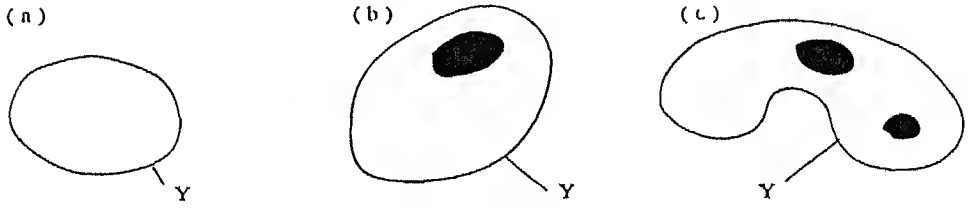


Figure 3.5 The three possible inclusion relations between Y and $Y \cap X_{h_m}$

Since all the possibilities have been dicussed, we take the second set of our recursion as follows

$$X_{h+1} = min_{h_m+1} \cup IZ_{I_{h+1}(I)}(X_h) \quad (3.6)$$

This relation holds for all levels of h. So the set of catchment basins of the gray scale image I is equal to the set X_{h_m} obtained from the following recursion

$$a)\ X_{h_m} = T_{h_m}(I),$$

$$b)\ \forall h \in [h_{min}, h_{max} - 1], X_{h+1} = min_{h+1} \cup IZ_{T_{h+1}(I)}$$

The watersheds of I correspond to the set of points of D_I, which do not belong to any catchment basin

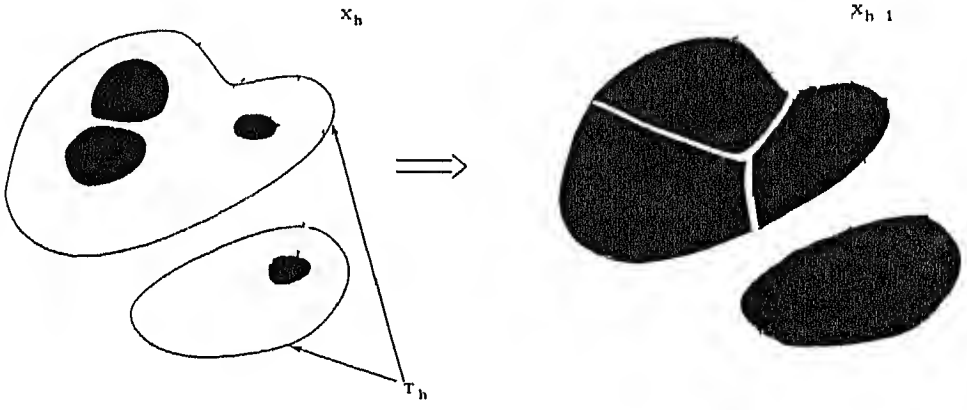


Figure 3.6 Recursion relation between X_h and X_{h+1}

3.4.2 Implementation of immersion algorithm

Implementation of immersion algorithm is divided into two parts

Sorting In order to have a direct access to pixels at a given level, the first step consists in an initial sorting of the pixels in the increasing order of their grey values

Flooding Once the pixels have been sorted, we proceed to the progressive flooding of the catchment basins of the image. Suppose the flooding has been done upto a

given level h Every catchment basin already discovered i e , every catchment basin whose corresponding minima has an altitude lower or equal to h is supposed to have a unique label Because of the initial sorting we can now access the pixels of altitude h+1 directly and given them a special value say MASK Those pixels among them which have an already labelled pixel as one of their neighbours are put into the queue Starting from these pixels the queue structure enables one to extend the labelled catchment basins inside the mask of pixels having value MASK, by computing geodesic influence zones After this step, only the minima at level h+1 have not been reached Indeed they are not connected to any of the labeled catchment basins Therefore a second scanning of the pixels at level h+1 is necessary to detect the pixels which still have a value MASK, and to give a new label to the discovered catchment basins The process continues until the highest gray level is scanned Only the pixels which are exactly "half way between" two catchment basins are treated as watershed pixels The algorithm for fast watersheds is given in Appendix A

In this way a meaningful segmentation of the frame is achieved At the end of this process we have the information about the number of regions identified and pixel label This information will be necessary throughout the coding process because we'll be dealing with the motion of segmented objects(regions) rather than individual pixels

Figures 3 7 show the result of segmentation of a particular frame in *CLAIRE* sequence and *SAILSMAN* sequence respectively

(a) (b)

Figure 3 7 Segmentation results for a) *CLAIRE* image and b) *SALESMAN* image

Chapter 4

MOTION ESTIMATION

Motion anlysis is an important step in image analysis The three main existing techniques for motion estimation can be categorised as feature based methods, that extract image features and track these features from frame to frame, secondly the gradient based methods that use spatial and temporal partial derivatives to estimate image flow at each location in the image and thirdly, the correlation based methods that use similarity in brightness patterns to determine motion vectors We choose the feature based approach as offers several advantages[7] Firstly, feature extraction reduces the vast amount of data present in an image, without necessarily eliminating salient information It can use either low level discriptors or high level discriptors as basic matching elements Low level approach is generally preferred for its generality and graceful degradation[7] Low level discriptors may be classified as either region based or edge based or point based The point features are most suitable as matching tokens In the motion estimation algorithm developed in this chapter we choose corner points as basic matching elements

4 1 Moving Region Selection

Moving regions are selected from the segmented regions of the previous decoded picture and the current input picture The regions in which the percentage of pixels which are different between the previous and the current frame, is larger than a threshold [MOV THset to 5 in our case] are marked as moving regions through a flag which is set to one A plot of the number of moving regions , as identified frame wise is shown in figure 6 4

4 2 Corner Detection

After the selection of the moving regions, We find the corner points in the current frame Corner point extraction is the basic step for motion estimation Figure 4 1 gives the total block diagram for Motion estimation A corner point is defined as a point where, both the image curvature and the gradient perpendicular to the edge, must have a local maxima and each exceed a specific threshold This is known as the Wang Brady criterion for corner detection [7]

The conditions for corner detection are as follows

$$C = \left(\frac{\partial^2 I}{\partial t^2}\right)^2 - S\|\nabla I\|^2 \rightarrow max \qquad \text{, where C is the curvature}$$
$$C > R$$
$$\|\nabla I\|^2 > E$$

The four parameters used in the above criterion can be understood as follows

S, a measure of the surface curvature ,

R a threshold for the corner response

E, an edge strength measure

M defining the window size over which the local maxima should be obtained

The values for these parameters in our simulation are

$$S = 0.4,\ R = 1e5,\ M = 5 \text{ (i.e. } 5 \times 5 \text{ window)}$$

Prior to applying the above criteria for corner detection, it is important to apply a smoothing filter over the image. This reduces noise reduction and suppresses of false corners. We have used a Gaussian window of MEAN = 0, and variance SIGMA =1, for smoothing. The results of corner point extraction for various images are shown in figure 6.5

4.3 Corner matching and Tracking

After corner detection, corner matching and tracking an important step in the estimation of motion parameters is carried out. Our objective is to match the corner points in two consecutive ie. the current and the previous frame, such that they can be used for motion estimation. Suppose the matcher recieves as input two grey scale images, I_1 and I_2, along with thier respective corners X_i and X_j respectively (where i= 0,1,2, n 1 and j= 0,1 ,2, n 1) Its task is to match X_i to X_j'. This is a the well known correspondence problem , with worst case enumeration as nn' possible matches. Constraints from physical world are usually imposed to reduce the computational load. They are

- Each corner may pair with only one other corner
- Small search neighbourhoods are utilised since corners do not move for between

consecutive frames(the high frame rate ensures small inter frame motion)

- Local image patch correlation is employed to verify matches

To match a corner point X_1 in I_1 all the corner points in I_2 lying in the search window centered at the position of X_1 are candidates for the match Correlation is then performed between the image region surrounding the corner X_1 and the image region surrounding each candidate corner in I_2 The winning candidate is one with the highest correlation value c_{max}, provided c_{max} exceeds a specific threshold To correlate an image region T with an image region P, each of size $(W \times W)$, the correlation measure is given by,

$$c = \frac{W^2 \sum_k \sum_l T_{i+k\,j+l} P_{i+k\,j+l} - \sum_k \sum_l P_{i+k\,j+l}}{\sqrt{W^2 \sum_k \sum_l T^2_{i+k\,j+l} - (\sum_k \sum_l T_{i+k\,j+l})^2}\ \sqrt{W^2 \sum_k \sum_l P^2_{i+k\,j+l} - (\sum_k \sum_l P_{i+k\,j+l})^2}}, \tag{4 1}$$

where (i, j) indexes the centre of the pixel block and size W = 2w+1 pixel wide $T_{i,j}$ and $P_{i,j}$ are the respective intensity values, and k and l range from $-w$ to $+w$ inclusive

The matching proceeds as follows

Each corner point , while it lives, has the following information

a) Current position , and positions in the previous two frames

b) Frame numbers , telling when the point first occured and last occured

c) type of point (i e , strong match, Weak match or No match, from previous frame)

d) confidence of match(i e , correlation with the previous frame) This is referred to as corner history in the block diagram of Motion estimation, figure 4 1

- STEP1 Assuming the motion of the corner point is restricted to a 8×8 window, we find out those corner points in the previous frame(with sorrounding region of size 5×5) which have a correlation with the corner points in the current frame higher than a threshold MIN_CORR(= 0.6 in our case)

- STEP2 Now repeat step 1, for corner points from the current frame to those in the previous frame

- STEP3 All those points, which have a two way match are labelled as STRONG matches and are stored in a list of corner points along with all the required information(see above) The points in current frame, which have no match in the previous frame are labelled as New corner points and stored in the list The points in previous frame, which have no match in the current frame, can be classified as either vanished for a short time, or permanently disappeared To process these, we go to step 4

- STEP4 If we assume that the point in previous frame has vanished only in the current frame, then we can predict its motion from the last two values and use that to find a pseudo corner point for it in the current frame, i.e based on prediction, we try to locate a point in the current frame which has a correlation value higher than a threshold with the corner point in the previous frame We choose a predictor based on constant velocity model The finite difference form for this predictor is

  $X(k+1) = 2X(k) - X(k-1)$

  If such a pseudo corner point is found, then it is labelled as WEAK match and stored in the list If no such point can be found for the corner, then it is labelled as "disappeared" If the point has remained disappeared over the last two frames then it is assumed "dead" and is deleted from the list of corner points

The above tracking of corner points is done for each new frame It results, at any particular instant in a set of corner points of various types The number of corner points increases for the first few frames, but then stabilizes, as the death rate of the points catches up with the birth rate

4 4 Motion Estimation and Prediction

We employ a 3 D model for motion estimation and prediction It is based on the following hypothesis

1 Moving regions form a plane in the image plane

2 The motion is represented by a 3 D linear transformation

3 The projection from 3 D space to 2 D image plane is a central projection

Each segmented region can be represented as a planar object given by

$$ax+by+cz=1$$

Any kind of motion can be represented by a combination of a rotation vector R and a translation vector V

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix}$$

The relation between 3 D coordinates (x, y, z) and 2 D coordinates (X, Y) is described by central projection as

$$X = F\frac{x}{z} \qquad Y = F\frac{y}{z}$$

where F is the distance from the centre of projection to the image plane

Using the above expressions , the motion on the image plane $(X\ Y) \rightarrow (X', Y')$ is

described as

$$\left(X', \ Y' \right) = \left(\frac{a_1 X + a_2 Y + a_3}{a_7 X + a_8 Y + 1}, \ \frac{a_4 X + a_5 Y + a_6}{a_7 X + a_8 Y + 1} \right), \qquad (4.2)$$

where,

$$a_1 = \frac{R_{11} + aI_1}{R_{33} + cI_3} \quad a_2 = \frac{R_{12} + bV_1}{R_{33} + cV_3} \quad a_3 = \frac{(R_{13} + cV_1)F}{R_{33} + cV_3}$$

$$a_4 = \frac{R_{21} + aV_2}{R_{33} + cV_3} \quad a_5 = \frac{R_{22} + bV_2}{R_{33} + cV_3} \quad a_6 = \frac{(R_{23} + cV_2)F}{R_{33} + cV_3}$$

$$a_7 = \frac{R_{31} + aI_3}{(R_{33} + cV_3)F} \quad a_8 = \frac{R_{32} + bV_3}{(R_{33} + cV_3)F}$$

The eight variables $a_1 \sim a_8$ are motion parameters. Here (X, Y) indicate a source position in the current frame, and (X', Y') refers to a source position in the previous frame. The eight parameter set $(a_1 - a_8)$ defines a motion parameter and characterizes the motion of a region.

4.4.1 Motion Estimation

We need to estimate the motion parameters for each region, from the previous decoded frame and the current frame. This is done by using a least square fit of the representative points of the region, in the two frames. Each region, at any instant, is represented by its corner points(assuming a rigid 3-D body). Hence these points are used for the least square fit.

First of all, we estimate the motion for only those regions which have been previously declared as moving regions. This saves a lot of computation especially in the case of an head and shoulder image as both display the same motion, and rest of

the back ground remains static Next, if a region has many corner points, then we put a maximum bound on the number of points to be used for motion estimation, and moreover, choose only the points with maximum confidance values (priority is given to STRONG matches)

The block diagram of the motion estimation process is shown in Fig 4 1 Now that we have got our points for each moving region, we estimate the motion

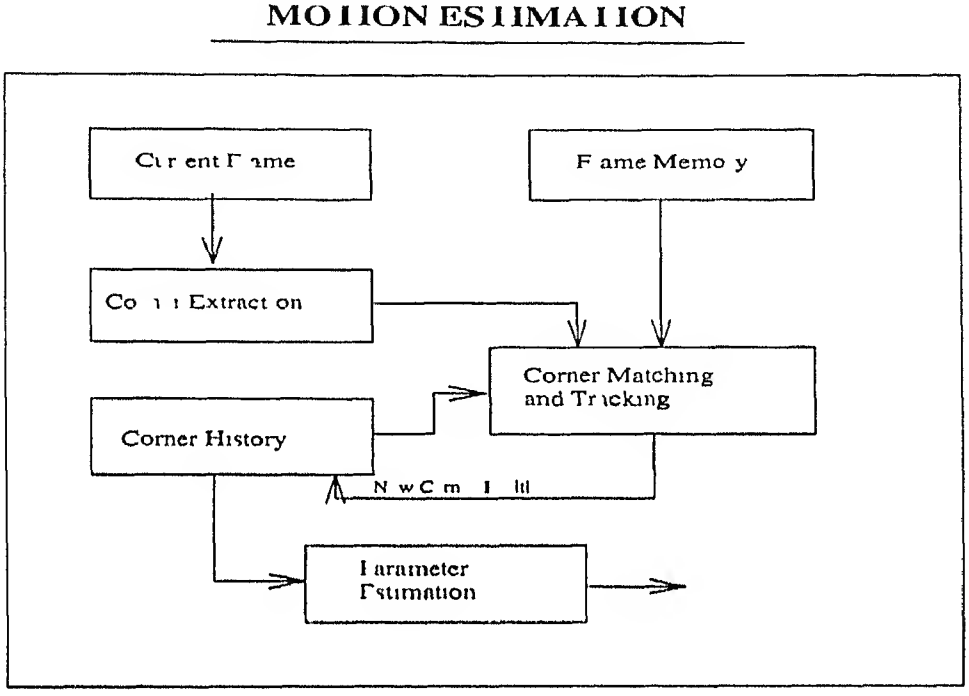


Figure 4 1 Block diagram for Motion Estimation

parameters (given above) as follows

Let X, Y and X', Y' represent the pixel position, in the previous and the current frame respectively Then the two coordinate sets are related by Eq(4 2) The error e_i^2 function, at a pixel, is defined as

$$e_i^2 = \left[X_i' - \frac{a_1 X_i + a_2 Y_i + a_3}{a_7 X_i + a_8 Y_i + 1}\right]^2 + \left[Y_i - \frac{a_4 X_i + a_5 Y_i + a_6}{a_7 X_i + a_8 Y_i + 1}\right]^2 \qquad (4\,3)$$

The cumulative Error function is given by

$$E = \sum_{i=0}^{n} e_i^2 (a_7 X_i + a_8 Y_i + 1)^2$$

Here the individual pixel errors have been multiplied by weights i e the denominator term of Eq (4 3) because it enables us to convert the non linear equation to a linear one thus permitting a simpler solution We have assumed that these weights not vary much in a segment such that the solution will be close enough to the optimum least square one

Let L be the matrix

$$L = \begin{bmatrix} X_1 & Y_1 & 1 & 0 & 0 & 0 & -X_1'X_1 & -X_1'Y_1 \\ 0 & 0 & 0 & X_1 & Y_1 & 1 & -Y_1'X_1 & -Y_1'Y_1 \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ X_n & Y_n & 1 & 0 & 0 & 0 & -X_n'X_n & -X_n'Y_n \\ 0 & 0 & 0 & X_n & Y_n & 1 & -Y_n'X_n & -Y_n'Y_n \end{bmatrix}$$

and A and B be defined as below

$$A^T = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 \end{bmatrix}$$

$$B^T = \begin{bmatrix} X_1' & Y_1' & & & & & X_n' & Y_n' \end{bmatrix}$$

Then the error function E can be expressed in terms of these matrices as follows

$$E = \|LA - B\|^2$$

In order to minimize E as a function of A , we compute the Pseudo Inverse of L , which is defined as follows

If the number of points is greater than 3 , then the Pseudo Inverse is

$$L^+ = (L^T L)^{-1} L^T$$

and we get a unique solution given by

$$A = L^{+}B$$

Otherwise if the number of points is less than 4 then Pseudo Inverse is

$$L^{+} = L^{I}(LL^{I})^{-1}$$

In this case, multiple solutions exist and the minimum norm solution is given by

$$A = L^{+}B$$

Since, in this case we would like the motion parameters to be closest to the " No Motion " case i e $a_1 = a_5 = 1$ and rest zero, so we substitute $a_1' = a_1 - 1$ and $a_5' = a_5 - 1$ Hence L remains same and B^I is replaced by

$$\begin{bmatrix} X_1' - X_1 & Y_1' - Y_1 & \cdots & X_n' - X_n & Y_n' - Y_n \end{bmatrix}$$

In this way we calculate $a_1 \sim a_8$ for each moving region

4 4 2 Motion compensated Prediction

The predicted picture is constructed from Eq (4 2) using the estimated motion parameters for each region Because the pixel location calculated from above is fractionally precise, the pixel value located between sample points is calculated by bilinear interpolation

Uncovered and overlapped regions are caused because the prediction is based on regions segmented at the previous picture The displacement vectors for the pixels in these regions are interpolated from adjacent motion parameters according to the distance from the region boundaries[1] The difference between the current input frame and predicted frame is called prediction error Coding of prediction error is explained in next chapter

Chapter 5

PREDICTION ERROR CODING

5 1 Intı oductıon

Predıctıon erroı ıs defined as the dıfference between the current ınput fı ame and a predıcted fraınc Referıng to the encodeı gıveı ın Fıgure 2 1, ıt ıs observed that thıs error typıcally arıses ın a head aıd shouldeı ımage because of the followıng reasoıs

1 The motıoı around the regıoıs lıke eyes, mouth etc ın a typıcal head and shoul der ımage cannot be sufficıently represeıted by motıoı parameters Consequently errors may be generated ın these regıoıs

2 Large errors may be generated aıound the segmented regıoı boundarıes due to regıoıs uncoveıed by the motıoı compeıs'ated pıedictıoı aıd maccuracıes ın the motıoı parameteı estımatıoı

So most of the pıedicted error ıs concentrated only around the face and boundarıes of movıng regıoıs Takıng caıe of these erroıs ıs very ımportant for exact segmentatıoı of the neıt fı ame and faıthful reconstı uctıoı of the orıgınal frame at the decoder

Coding the error image by standard methods like JPEG is not suitable for very low bit rate applications because this error information itself requires 48kbits for its representation Since most of the error is concentrated around the eyes and mouth and the boundaries of moving regions, there is no need to code the total error image So we adopted a new approach to encode the error picture by taking advantage of the fact that only few 8×8 blocks of the error image have significant error and the rest of the blocks are error free

5 2 Method used

Prediction error coding algorithm first divides the total error image into 8×8 blocks After that it operates on a block by block basis In the present approach we are not coding every 8×8 block in the error image The significant blocks in which the total absolute error is greater than a threshold (say T1) are counted as significant error blocks and coded

A two dimensional *DCT* is performed on each significant error block The magnitude of each *DCT* coefficient indicates the contribution of a particular combination of horizontal and vertical spatial frequencies to the significant error block The coefficient corresponding to zero horizontal and vertical frequency is called DC coefficient The *DCT* output data is an 8×8 block of transform coefficients with the DC term at the top upper left corner The other 63 coefficients are AC terms

DCT does not directly reduce the number of bits required to represent the significant error block For an 8×8 block of pixels, *DCT* produces an 8×8 block of coefficients The reduction in the number of bits follows from the observation that, for typical blocks the distribution of coefficients is non uniform The transform

tends to concentrate the energy into the low frequency coefficients and many of other coefficients are near zero The bit rate reduction is achieved by not considering the near zero coefficients and by quantizing and coding the remaining coefficients as described below The non uniform coefficient distribution is a result of the spatial redundancy present in the original image block

By experimentation it is found that a threshold of T1=800 is a reasonable value to select significant error blocks in an error image The label Block_Run is used to represent the number of blocks passed before encountering the significant error block As only significant error blocks are coded, it reduces the computational time as well as the total number of bits needed to encode This is very important for a low bit rate coder If the number of significant error blocks exceeds a threshold T2 (20 in our case) then there is a definite need to send an intra frame In that case we send the total image coded using standard JPEG method

5 2 1 Quantizing Scheme

For quantization, we use the standard *JPEG* quantizing scheme There is always a trade off between picture quality and degree of quantization A large quantization step size can produce unacceptably large image distortion Unfortunately, finer quantization leads to lower compression ratios The question arises how best one can quantize the *DCT* coefficients most efficiently As human visual system response has natural high frequency roll off, these frequencies play a less important role than low frequencies This allows one to use a high quantization step size for the high frequency coefficients, with little noticeable image distortion

The quantization matrix is a 8×8 matrix of different step sizes, corresponding to each *DCT* coefficient Step sizes will be smaller in the upper left part(low

frequencies) and large in the upper right part (high frequencies) The quantizer divides the *DCT* coefficients by its corresponding quantum then rounds it to the nearest integer Large quantums drive small coefficients to zero The result is many high frequency coefficients become zero and therefore easier to code The low frequency coefficients undergo only minor adjustment

5 2 2 Compression Technique

As mentioned earlier, in the present method we are not coding each and every block in the error picture We are coding only significant error blocks which have sufficient error The number of blocks (without sufficient error) passed before encountering the significant error block is the Block Run So each significant error block is associated with a Block Run Subsequently we apply *DCT* and quantization to each significant error block followed by Zig_Zag scanning of the quantized *DCT* coefficients and coding

Zigzag Scanning

After the quantization of each significant error block, it is not unusual for more than half of the *DCT* coefficients to be equal to zero Since most of the non zero *DCT* coefficients will typically be concentrated in the upper left hand corner of the matrix, it is apparent that a zig/zag scanning pattern will tend to maximize the probability of achieving long runs of consecutive zero coefficients Zigzag scanning pattern is shown in the figure 6 10 This ordering puts the lowest frequency coefficients first followed by the high frequency coefficients which typically have close to zero values

Each significant error block is associated with a Block_Run as previously mentioned This Block_Run is uniformly quantized and encoded The dc and ac terms in each error block are then separately coded as in standard JPEG coding

Block_Run is uniformly quantized and encoded using fixed length coding We allocated 8 bits to represent this information As regards the DC coefficient encoding, the absolute values of dc coefficients are not directly coded This is because the DC coefficients of neighbouring blocks typically do not vary Consequently fewer bits are needed to encode the differences between the successive DC coefficients rather than the absolute values So the difference between the DC coefficients is encoded DC coefficient difference is encoded into two parts, some what like exponent and mantissa One part is the magnitude category The magnitude category M of an integer N is the smallest integer M such that
$2^M > |N|$

The magnitude category M is Huffman coded and M bits appended to it to identify a particular member of the category For example if the difference is 5, it is in category 3, which comprises the set { 7, 6, 5, 4, 4, 5, 6, 7} Now this category 3 is Huffman coded as 100 using standard JPEG dc coding tables and then additional 3 bits ie 101 are appended to it to represent a paticular number in the category So total code is 100101 to represent the DC coefficient

After coding the dc difference, the algorithm codes the block quantized AC coefficients The quantized ac coefficients are zig/zag scanned to form a sequence of (*Run*,*Size*) pairs Beginning at the upper left hand corner we scan the coefficients in zigzag fashion The number of zero values passed before encountering the non zero value is the Run count and the non/zero value is the *Level* The *Size* is the

magnitude category of *Level*. So the magnitude category *Size* of integer *Level* is the smallest integer *Size* such that

$$2^{Size} > |Level|$$

Now this (*Run*,*Size*) pair is Huffman coded using standard JPEG ac coding tables and then "*Size* number" of bits are appended to identify the particular member of the magnitude category *Size*. This procedure is repeated for all the significant error blocks in the error image.

Standard JPEG coding takes approximately 50Kbits to code the entire error image. Following the improved method of coding only the significant error blocks we achieved a compression of about 50:1 which is very important for low bit rate coding.

Chapter 6

RESULTS AND DISCUSSION

In this chapter we evaluate the performance of the Region based Low Bit Rate Video Codec developed The performance is tested using standard sequences such as 1)Claire sequence 2) Salesman and also with a sequence taken in the lab and titled as the "Author sequence' The size of the images used in all the sequences were 256 × 256 with a frame rate of 7 5 frames/sec The algorithm developed has successfully reconstructed the first 10 frames of Claire,Salesman and Author sequences The original and reconstructed frames are shown in figures 6 11 to 6 40

As discussed in the previous chapters, the low bit rate video codec essentially consists of the following stages

- Segmentation
- Motion estimation
- Coding

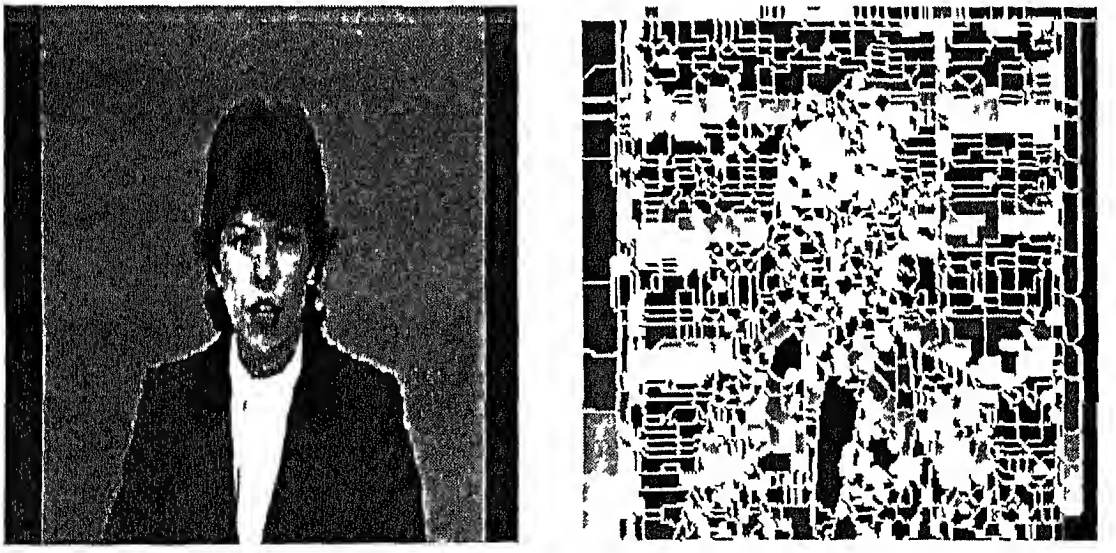
Using the standard sequences, we have been able to successfully segment each frame, identify corner points, and track them over the frames Using the corner points, we extract motion information and predict the next frame The predicted frame is then used to calculate the prediction error, which is then efficiently coded

At the decoder we use this error and the motion information to perform the reconstruction of the current frame

The various steps involved are briefly discussed from the implementation point of view and the results obtained are shown for a particular frame of each sequence

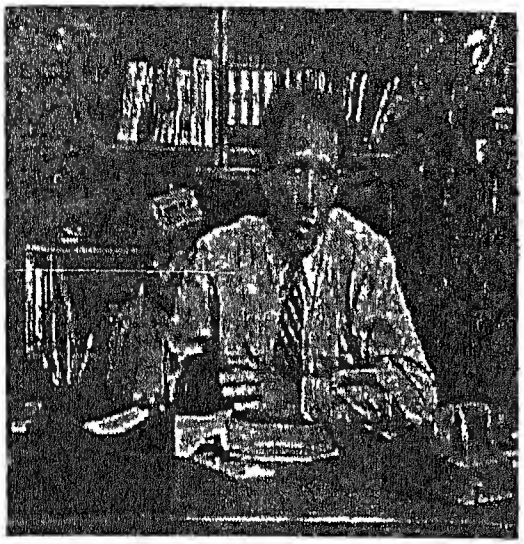
6 1 Segmentation

The Watershed transformation is explained in chapter 3 has been implemented for image segmentation The effectiveness of the image segmentation method based on watershed transformation is limited by the quality of the gradient image Watershed tranform based on conventional gradient operators results in over segmentation Because of this each homogeneous region is partitioned into a large number of regions and proper contours are lost in a multitude of false ones This can be observed from the segmentation results of the Claire and salesman images obtained using watershed algorithm based on conventional morphological gradient operator, shown in figures 6 1 and 6 2

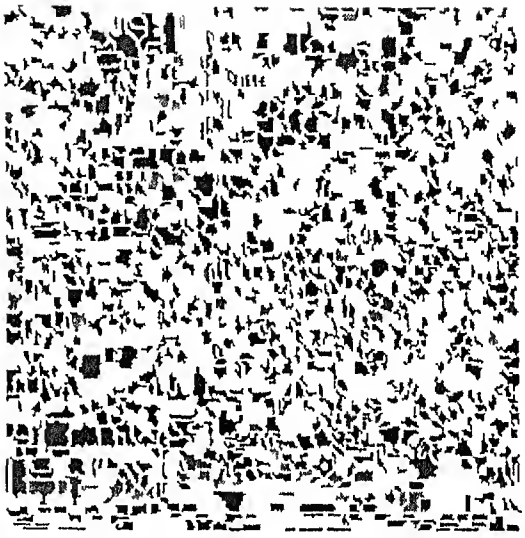


(a) *CLAIRE* Image (b) Over Segmented *CLAIRE*

Figure 6 1 Over Segmentation Due to Conventional Gradient Operator



(a) *SALESMAN* Image

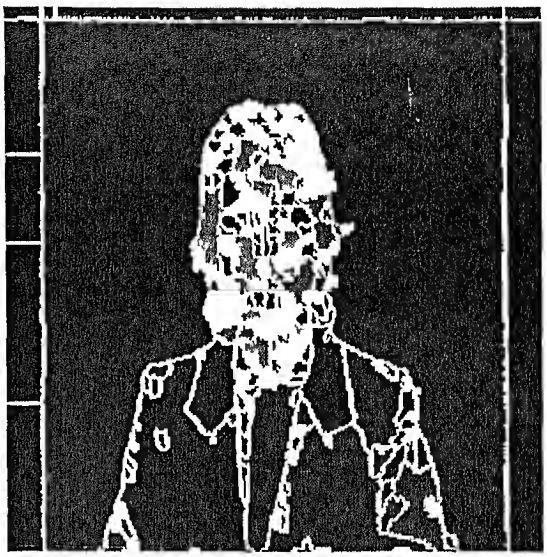


(b) Over Segmented *SALESMAN*

Figure 6.2 Watershed Segmentation with Conventional Gradient Operator

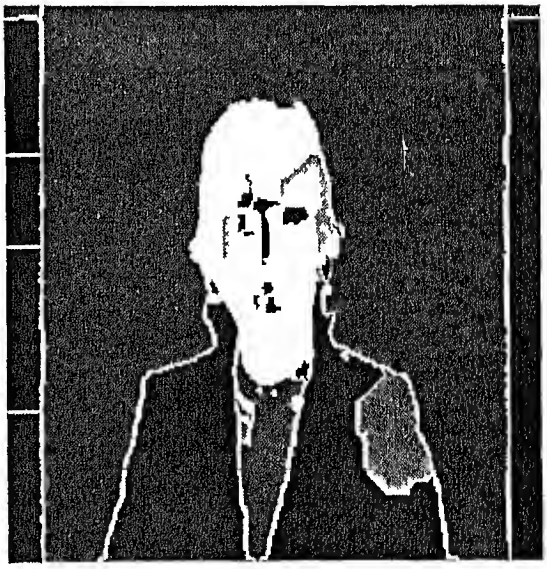
The oversegmentation is due to the local minima caused by noise and quantization error. By thresholding, it is not possible to eliminate these local minima while preserving those produced by blurred edges. Multi scale gradient algorithm discussed in section 3.2, has been employed to alleviate this problem by efficiently enhancing blurred edges. This enhancement increases the gradient value for blurred edges above those caused by noise and quantization error. Consequently the local minima produced by noise and quantization error are eliminated using the algoritm discussed in section 3.3. In this procedure, a constant 'h' is added to the dilated gradient image and local minima with a contrast lower than h is filled using the reconstruction by erosion operation as discussed in section 3.3. The parameter 'h' controls the number of segmented regions. As 'h' increases, the number of regions produced decreases. Figure 6.3 shows the effect of varying 'h' on the segmentation process.

(a) *CIAIRL* Image



(b) Segmentation with h=15

(c) Segmentation with h=25



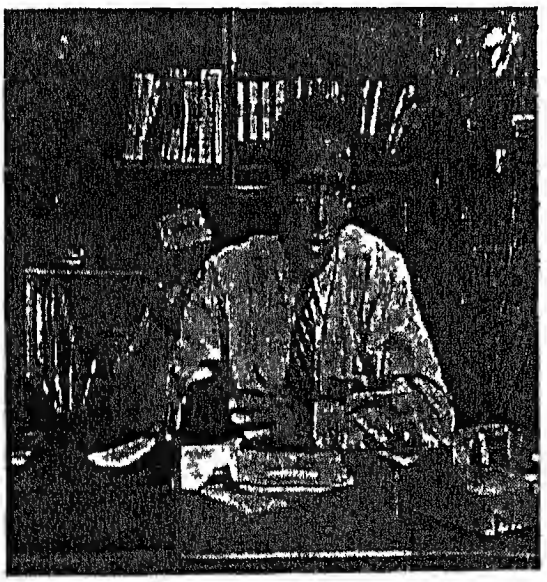
(d) Segmentation with h=40

Figure 6.3 Variation of Segmentation with parameter 'h' In the Multiscale Gradient

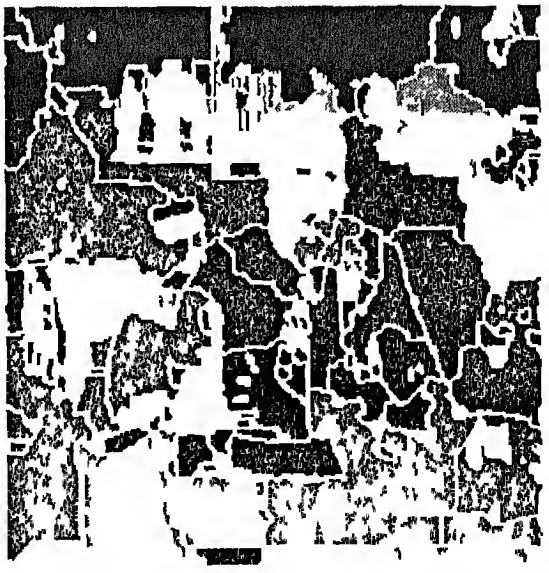
Watershed transformation using Multiscale gradient algorithm results in a meaningful segmentation of a frame without requiring any further region merging. This is observed from the following segmentation results of *CLAIRE* and *SALESMAN* image shown in fig 6.4

(a) *CLAIRE* Image

(b) Segmented *CLAIRE* Image



(a) *SALESMAN* Image



(b) Segmented *SALESMAN* Image

Figure 6.4 Watershed Segmentation with Multiscale Gradient

6 2 Motion Estimation

The Motion estimation procedure explained in chapter 4 has been implemented After segmentation moving regions are selected from the segmented regions explained in section 4 1 A plot of the number of moving regions is identified frame wise is shown in fig 6 5

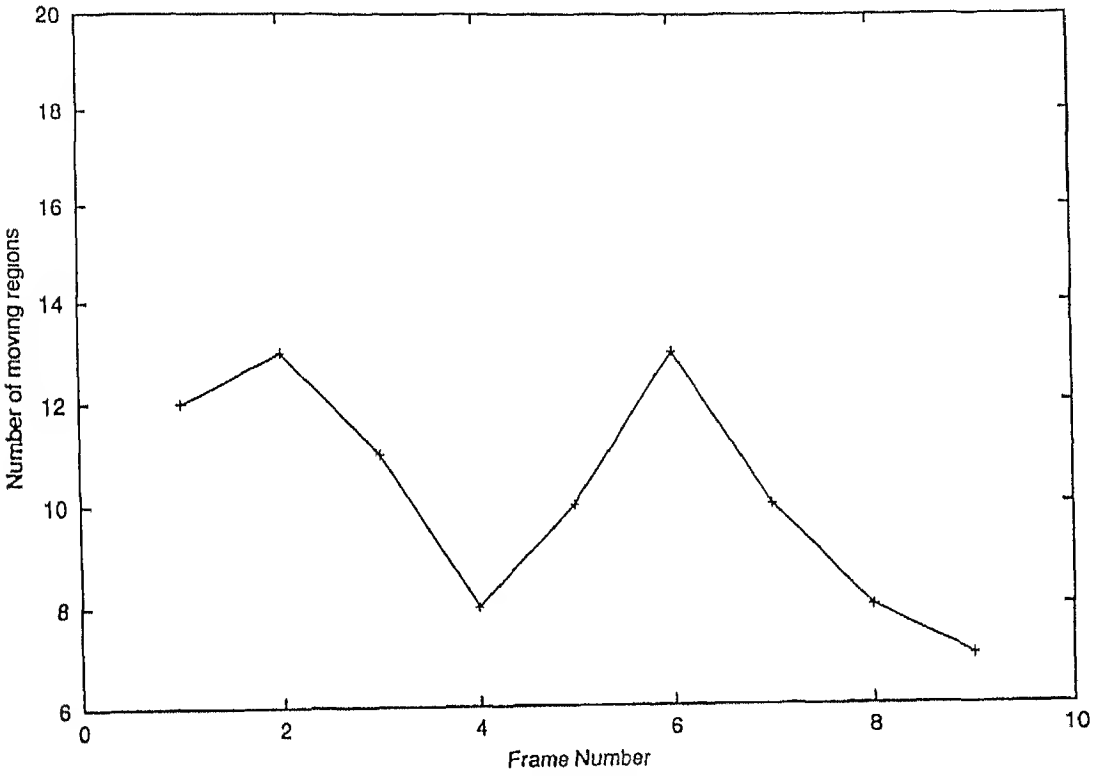


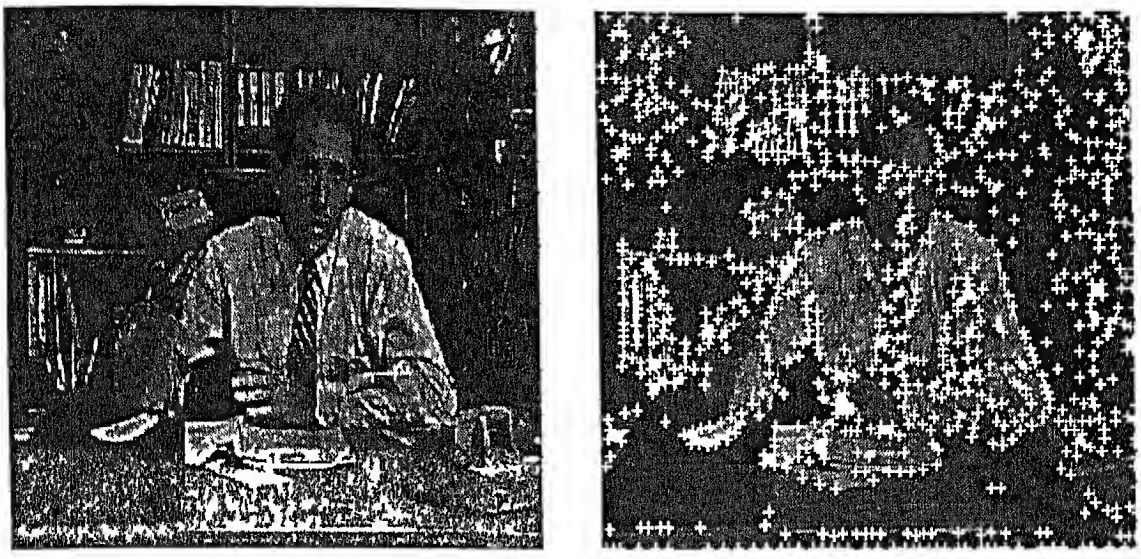
Figure 6 5 Number of Moving Regions against Frame Number

After selection of moving regions corner points are identified in each frame The corner point extraction method explained in section 4 2 is implemented The four parameters to be specified in the corner detection are S R D and M The mask size M controls the extent over which the corner position must be a local peak and indirectly defines the minimum separability between corners The larger mask rejects the high frequency intensity variations (noise) and reduces clutter A mask size of 5

proves satisfactory for our application The parameter S supresses the corner corner response along edges by subtracting a fraction of the edge strength Increasing S beyond 0 3 0 4 has little effect on edge response suppression, since any corners remaining along the edges must arise from significant surface curvature changes However as S tends to infinity useful corners begin to disappear A balance must be struck between S and R S must be sufficiently large to cancel out spurious edge response, but beyond that, it simply introduces an ' offset" which R can compensate for We limit S to the value 0 4, and set R = 0 A value of E = 10000 works well Figure 6 6 shows the results of corner point extraction for the *CLAIRE* and *SAI I SMAN* sequences respectively

(a) *CLAIRI* Image

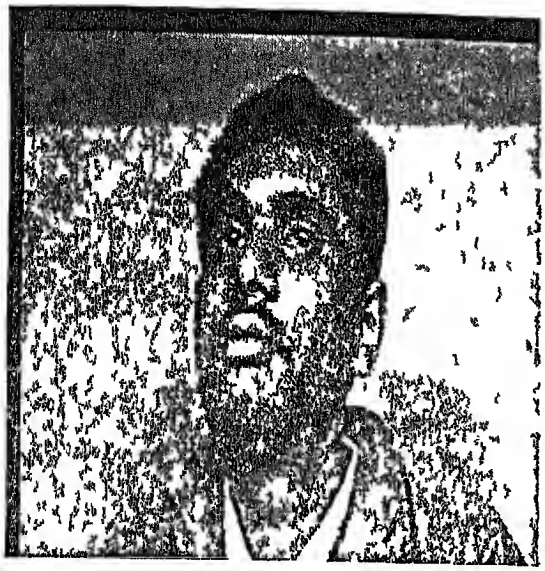
(b) Corner Detection in *CLAIRE*



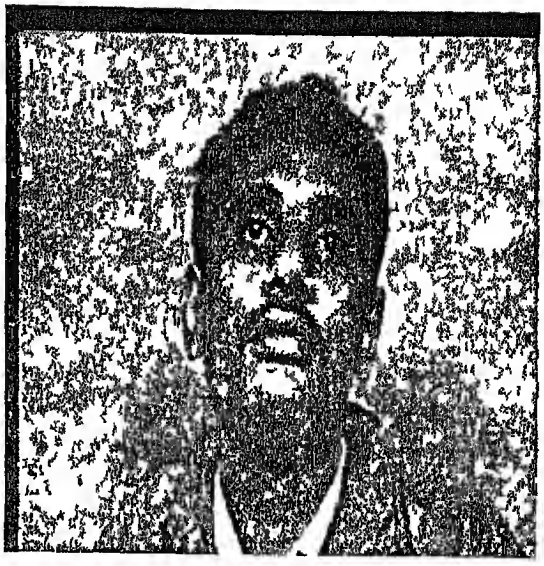
(a) *SALESMAN* Image (b) Corner Detection In *SALESMAN*

Figure 6.6 Corner detection in a frame

After finding the corner points of current and previous frames, our objective is to match the corner points in the two frames. The procedure for corner tracking and matching is explained in section 4.3 is implemented. In matching the corner points of previous frame with the current frame the winning candidate is one with the highest correlation value c_{max} , provided c_{max} exceeds a specific threshold *MIN_CORR* in our case. This threshold is necessary since the "best" corner in the window need not be a valid match, e.g. the true future may have disappeared. In our case we set *MIN_CORR*=0.6. This worked well for *CLAIRE* sequence, but for *SALESMAN*, because of the outliers the least square method for calculating motion parameters did not converge. So we set *MIN_CORR* 0.8 in this case. But this is not a solution. A robust method for rejecting outliers is needed. Figure 6.7 show the corner point tracking from previous frame to current frame for a particular frame of the *AUTHOR* sequence.



(a) Current Frame



(b) Previous Frame

(c) Corner Tracking

Figure 6.7 Corner tracking for a Particular Frame

From the corner point tracking information motion parameters are extracted and used to predict the next frame The least squares method is used for estimating the motion parameters

6 3 Coding

Prediction error which is the difference between current frame and predicted frame is encoded in an efficient way suitable for low bit rate applications In addition motion parameters are encoded using fixed length coding

6 3 1 Prediction error coding

In Prediction error coding stage we first divide the error image into 8×8 blocks Subsequently we operate on a block by block basis In this method, not all error blocks are coded The significant error blocks in which the total absolute error is greater than a threshlod (say T1) are only transform coded using the *DCT* By experimentation, we found T1=800, as a reasonable value This reduces computational time and number of bits required to encode, as a result of increasing the efficiency of coder The number of blocks passed before encountering the significant error block is called Block_Run This is shown in figure 6 8 After moving part the two error blocks the first significant error block is encountered and so is assigned a Block Run of two Similarly, each significant error block is associated with a Block Run as shown in figure 6 8

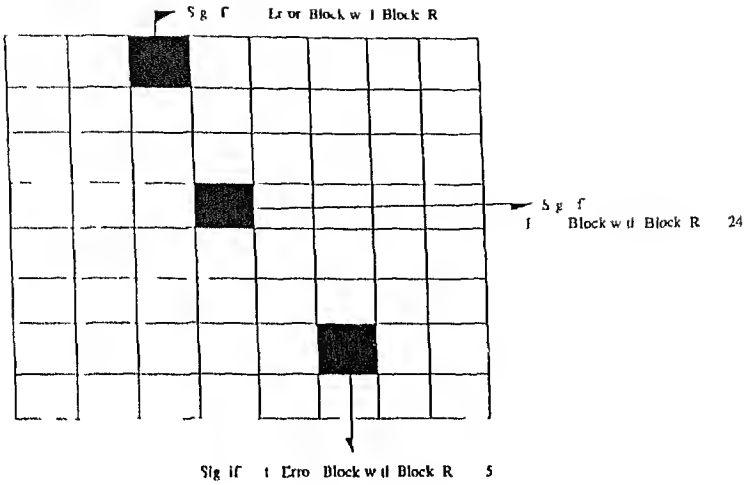


Figure 6 8 Error Blocks with Block Run

The followings figure 6 9 shows the original error image and modified error image with only significant error blocks

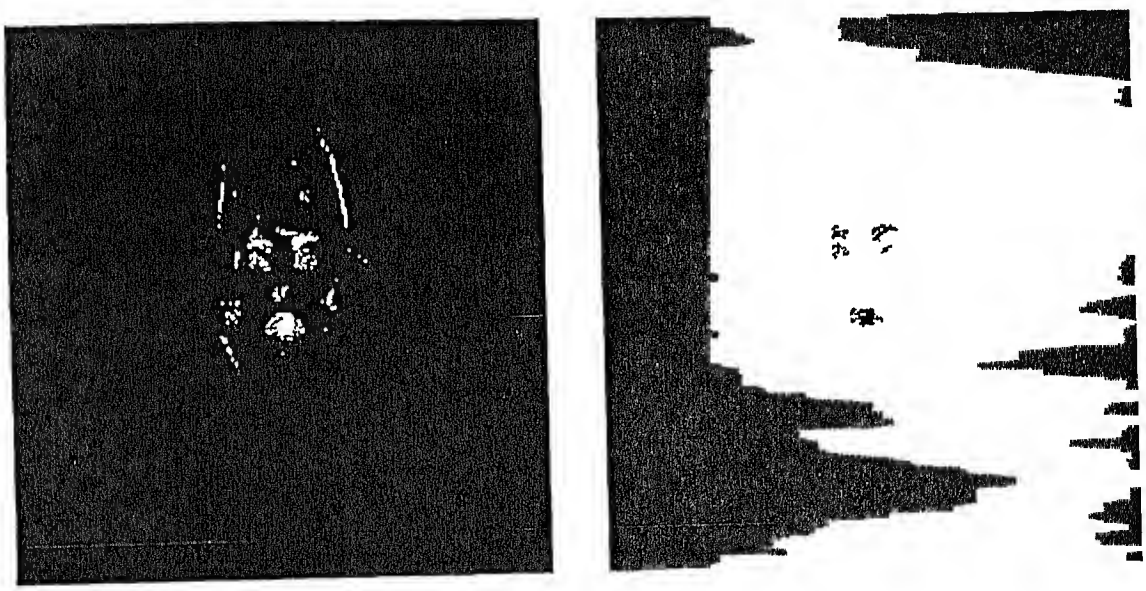
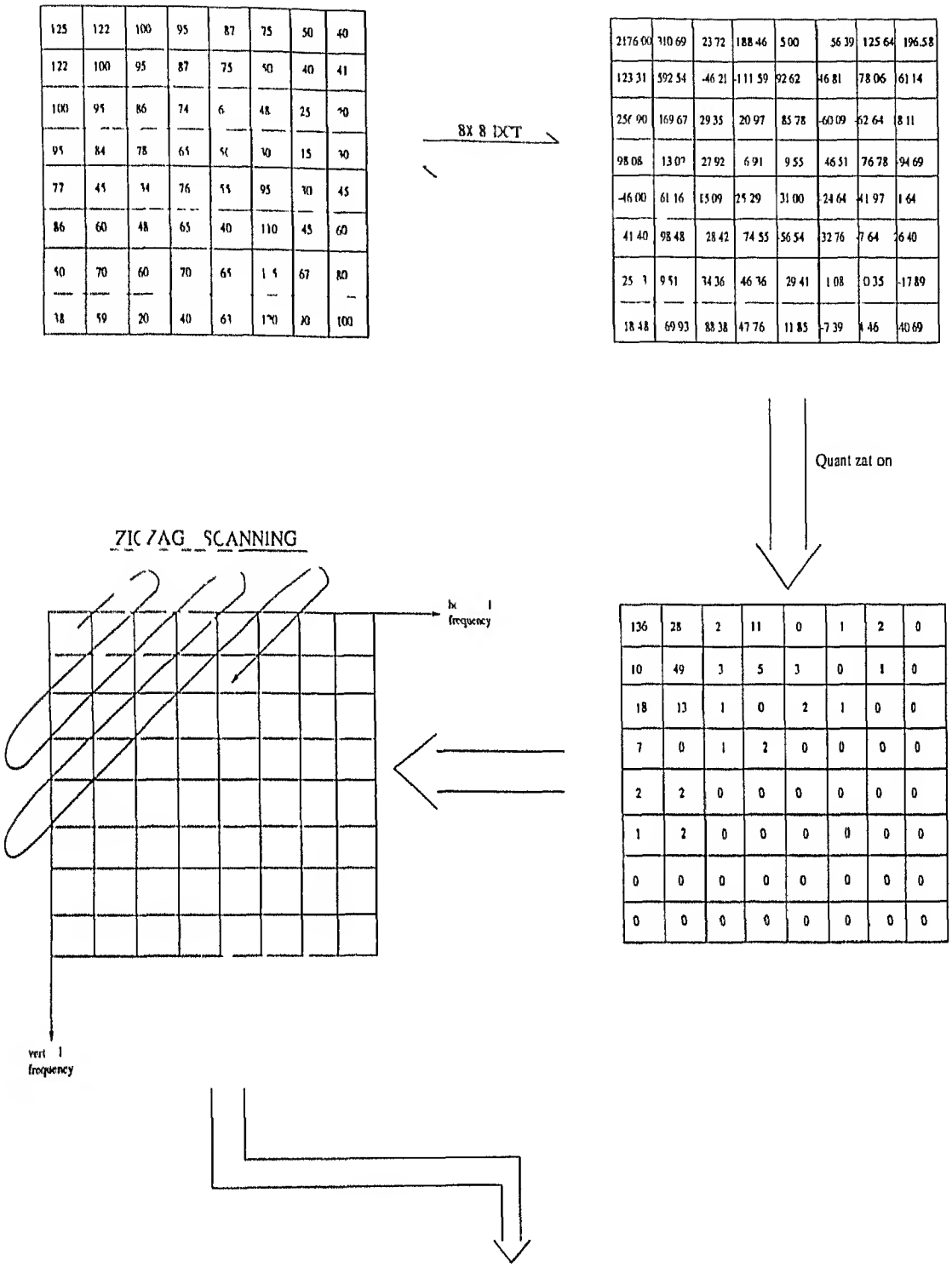


Figure 6 9 Original Error Image and Modified Error Image

Each significant error block is transform coded employing *Discrete Cosine Transform* followed by Zig/Zag scanning of the quantized *DC I* coefficients and coding as explained in sections 5 1 5 2 and 5 3 respectively The following figures explain this method diagrammatically with results



Run Level Pairs (0 5) (0 4) (0 5) (0 6) (0 2) (0 4) (0 2) (0 4) (0 3) (0 2) (1 1) (0 3) (1 1) (0 2) (1 1) (0 2) (0 1) (1 2) (1 2) (0 2) (1 2) (1 1) (0 1) EOB

Figure 6.10 Coding Method for a Error Block

6.3.2 Motion parameters coding

Motion parameters are encoded using uniform quantization and fixed length coding. We are allocating 8 bits for each motion parameter.

6.4 Estimated bit rate

Using the above technique for coding, we got an estimate of the bit rate for *CLAIRE* sequence as follows.

For each frame

Average number of regions = 50

Average number of moving regions = 12

Number of bits required to code each moving region = 48

Bits required for motion information = $12 \times 48 = 576$

Avg number of error blocks = 15

Average number of bits required to code error information = 1100

Flag bits for regions(moving/static) = 50

Hence, Number of bits/frame = 576+1100+50 = 1726

Frame rate = 7.5 frames/second

Therefore, **ESTIMATED BIT RATE** = 7.5×1726 = 13K(approximately)

6.5 Discussion

The results of simulation for standard sequences such as 1)*CLAIRE* 2)*SALESMAN* and also for the sequence taken in the lab titled as the "*AUTHOR* sequence" are

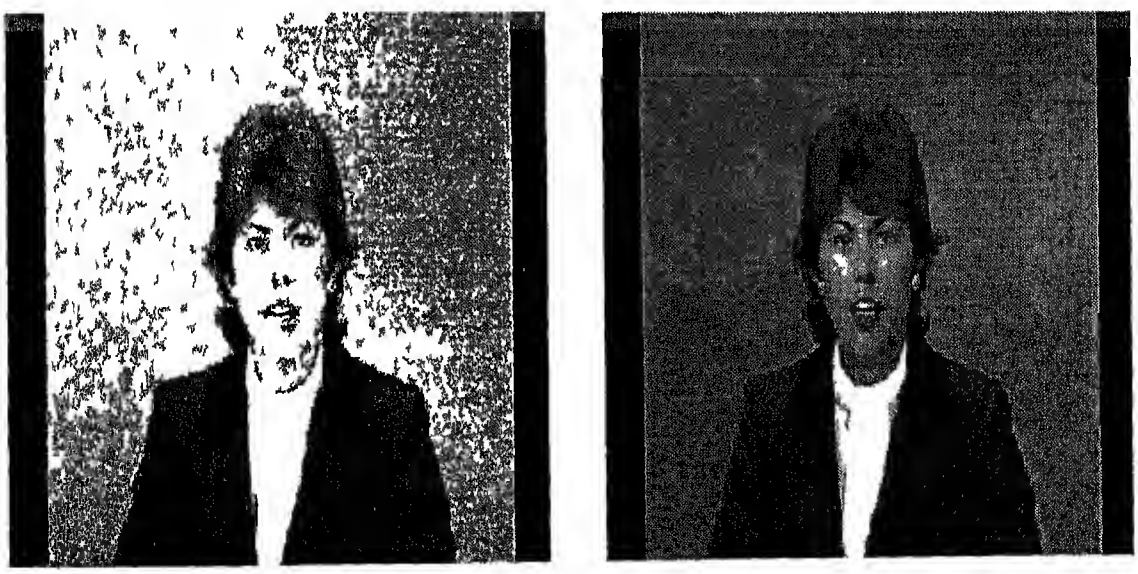
shown in figures 6 11 6 40 On the left is the original frame, and on right is the re constructed frame after motion compensation prediction in the decoder The PSNR obtained is 38 34 for different sequences indicated good quality of reconstructed images

Mean square error (MSE) and signal to noise ratio (SNR) variations for *CLAIRL* and *SAILSMAN* sequences are shown in figures 6 41 6 40 The Variations in MSE and SNR with frame number change according to the motion of objects in the corresponding frames If the Prediction is poor, then the prediction error will be more and as each time the previous decoded frame is segmented this error will accumulate over the frames Consequently SNR decreases with the number of frames The SNR also degrades when the motion between frames is large and include complex motion

Change in SNR with bit rate for standard *CLAIRE* and sales man sequences is shown in figures 6 45 and 6 46

(a) Original (b) Reconstructed

Figure 6.11 Original and reconstructed *CLAIRE* Sequence Frame 1



(a) Original (b) Reconstructed

Figure 6.12 Original and reconstructed *CLAIRE* Sequence Frame 2

(a) Original (b) Reconstructed

Figure 6.13 Original and Reconstructed *CLAIRE* Sequence Frame 3

(a) Original (b) Reconstructed

Figure 6.14 Original and Reconstructed *CLAIRE* Sequence Frame 4

(a) Original (b) Reconstructed

Figure 6.15 Original and Reconstructed *CLAIRE* Sequence Frame 5

(a) Original (b) Reconstructed

Figure 6.16 Original and Reconstructed *CLAIRE* Sequence Frame 6

(a) Original (b) Reconstructed

Figure 6.17 Original and Reconstructed *CLAIRE* Sequence Frame 7

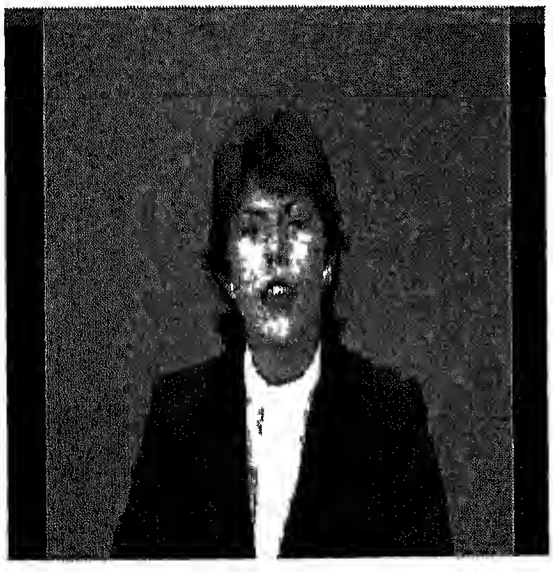
(a) Original (b) Reconstructed

Figure 6.18 Original and Reconstructed *CLAIRE* Sequence Frame 8

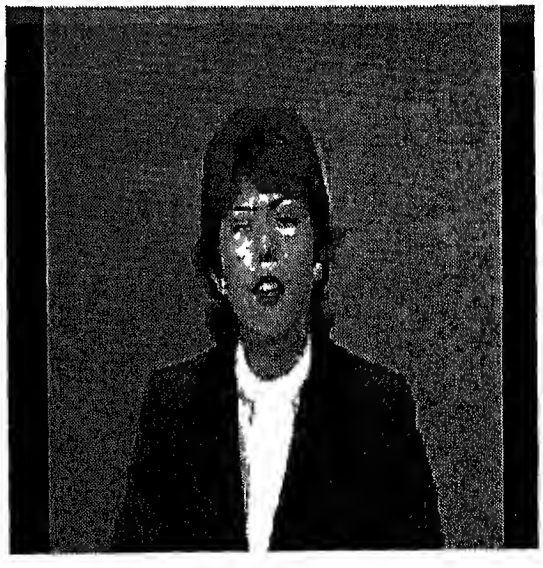
(a) Original

(b) Reconstructed

Figure 6.19 Original and Reconstructed *CLAIRE* Sequence Frame 9

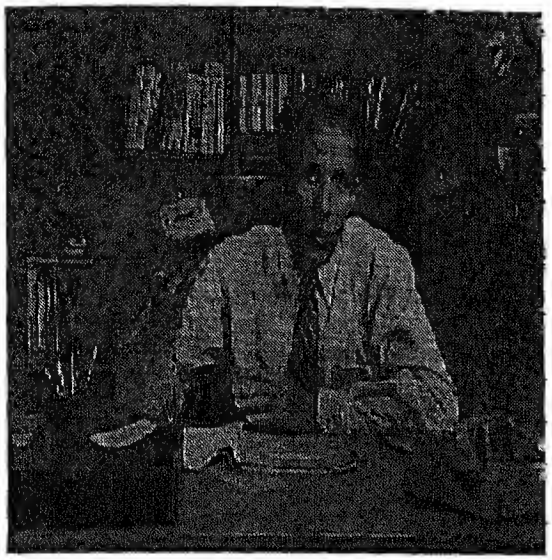


(a) Original

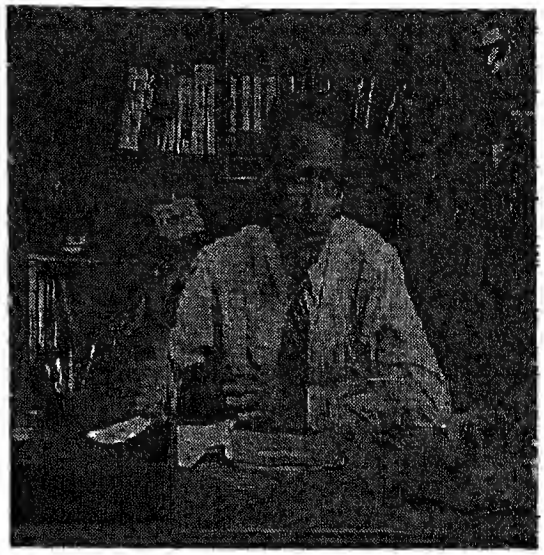


(b) Reconstructed

Figure 6.20 Original and Reconstructed *CLAIRE* Sequence Frame 10

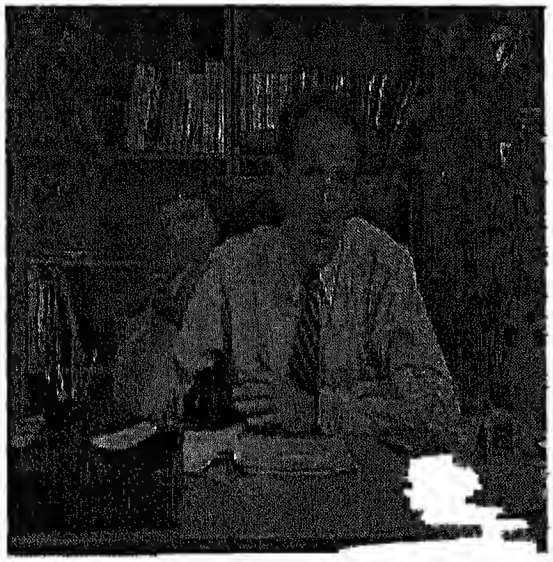


(a) Original



(b) Reconstructed

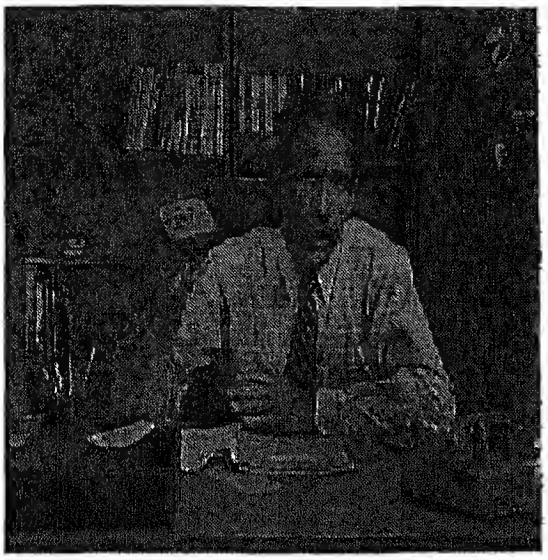
Figure 6.21 Original and Reconstructed *SALESMAN* Sequence Frame 1



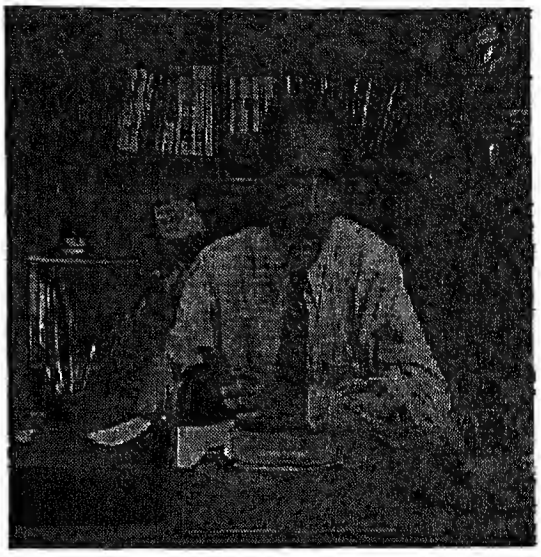
(a) Original

(b) Reconstructed

Figure 6.22 Original and Reconstructed *SALESMAN* Sequence Frame 2

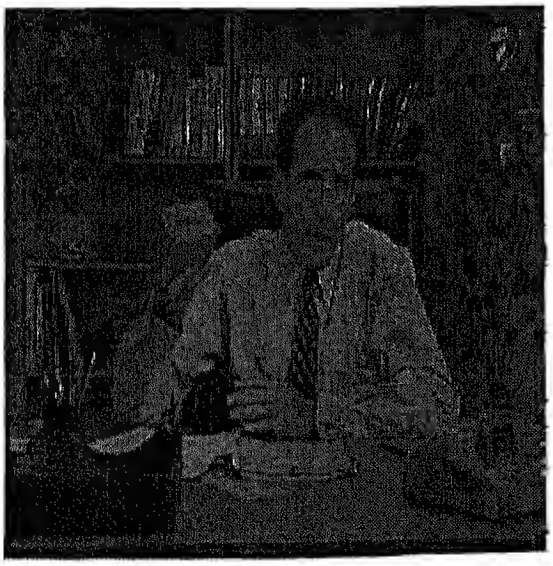


(a) Original

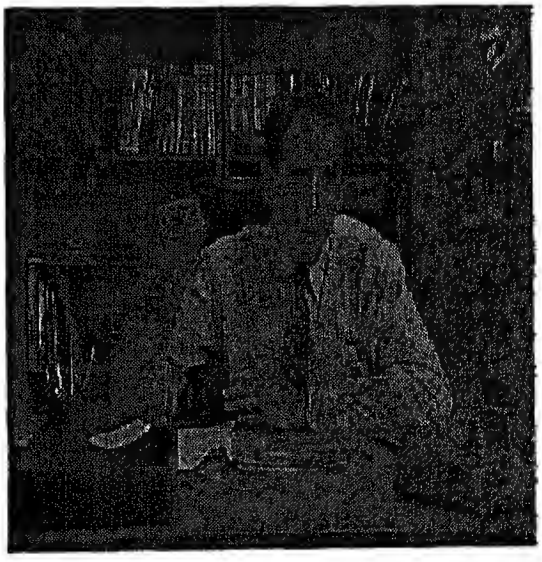


(b) Reconstructed

Figure 6 23 Original and Reconstructed *SALESMAN* Sequence Frame 3

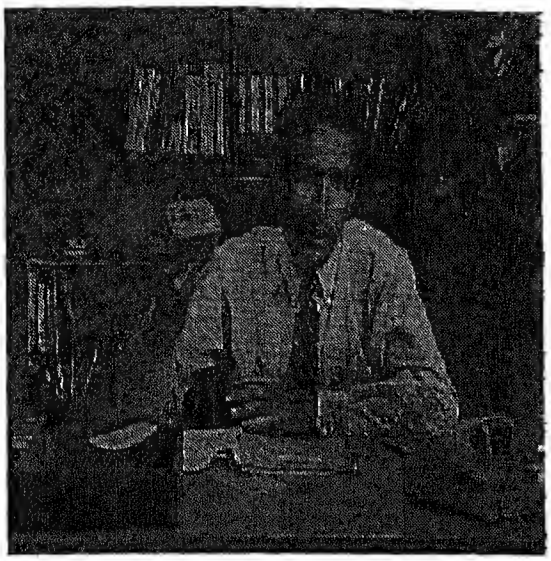


(a) Original

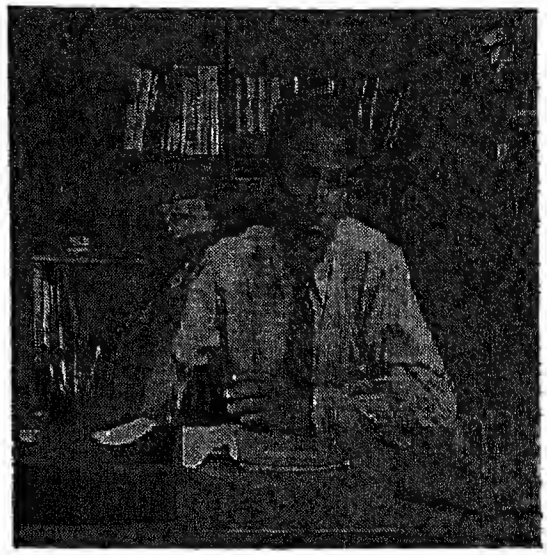


(b) Reconstructed

Figure 6 24 Original and Reconstructed *SALESMAN* Sequence Frame 4

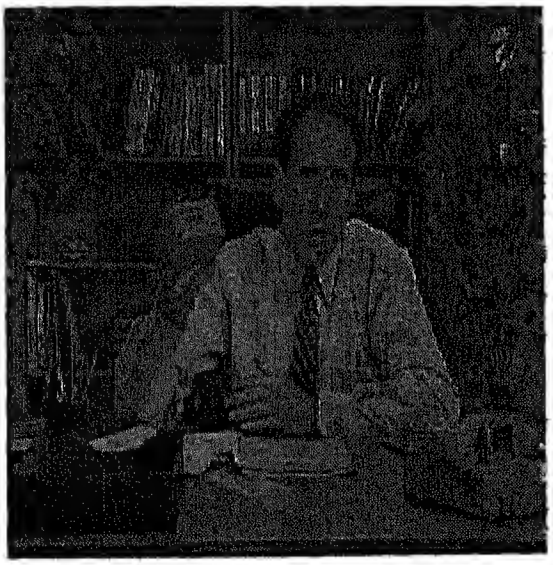


(a) Original



(b) Reconstructed

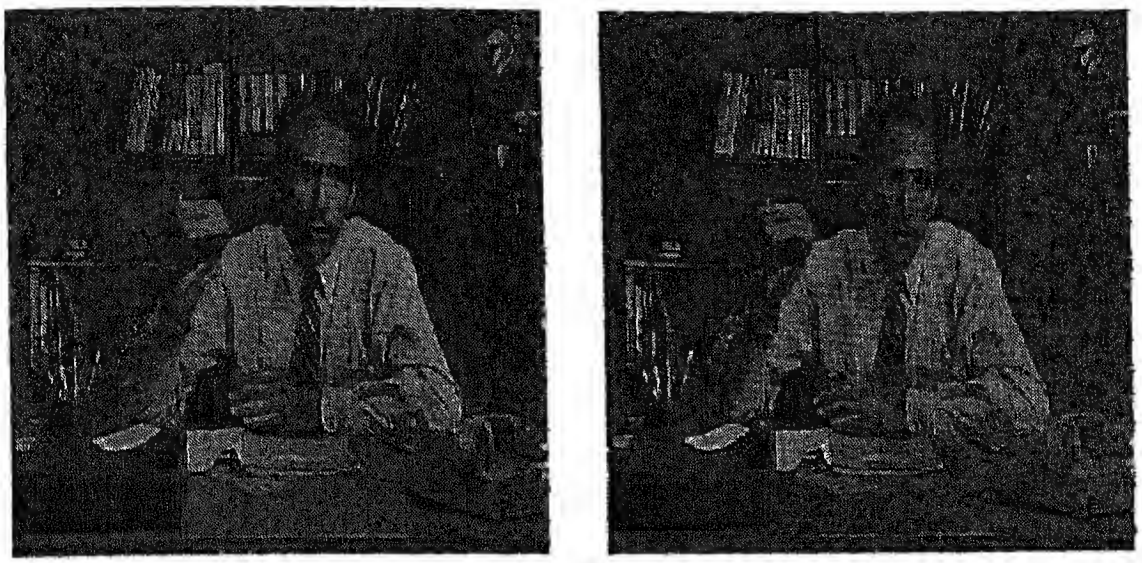
Figure 6 25 Original and Reconstructed *SALESMAN* Sequence Frame 5



(a) Original

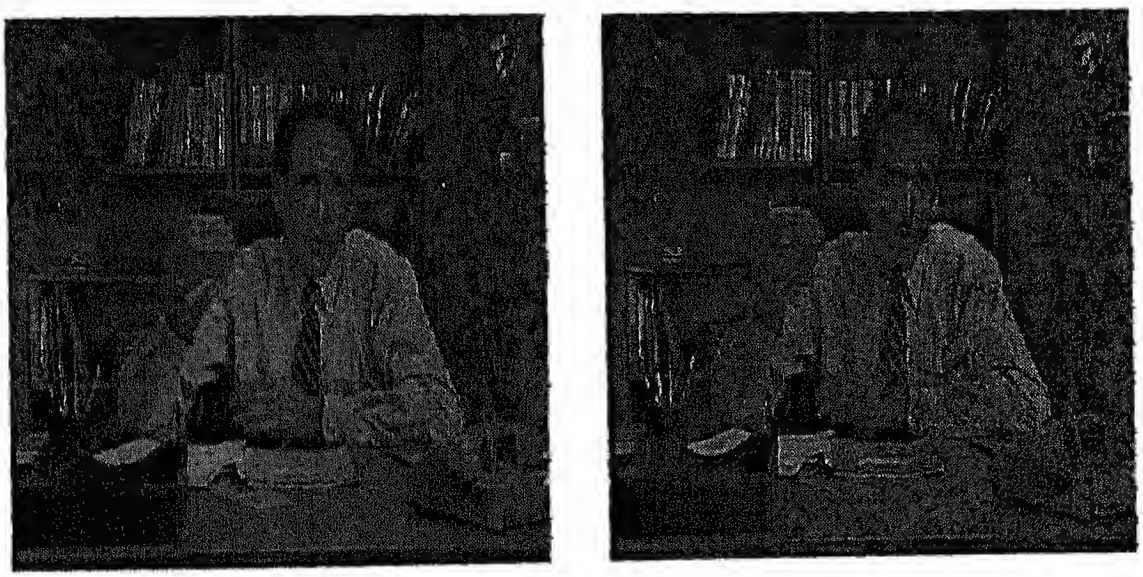
(b) Reconstructed

Figure 6 26 Original and Reconstructed *SALESMAN* Sequence Frame 6



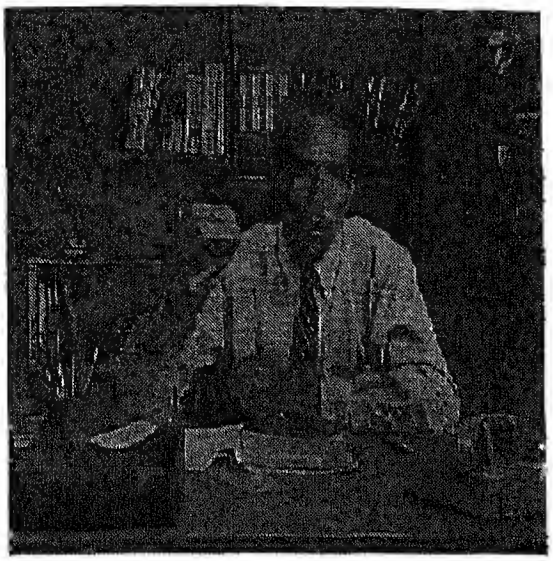
(a) Original (b) Reconstructed

Figure 6.27 Original and Reconstructed *SALESMAN* Sequence Frame-7

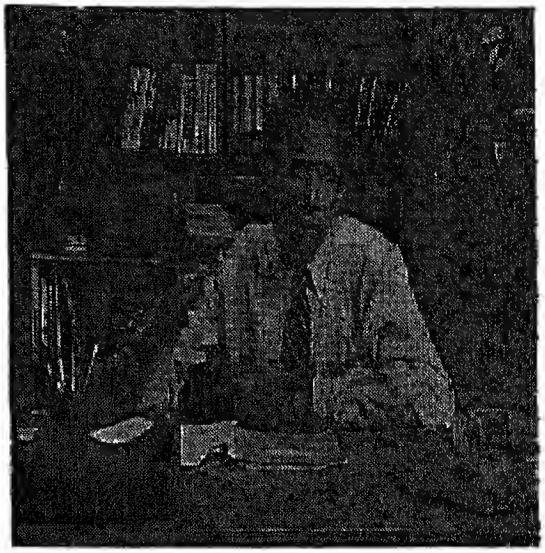


(a) Original (b) Reconstructed

Figure 6.28 Original and Reconstructed *SALESMAN* Sequence Frame-8

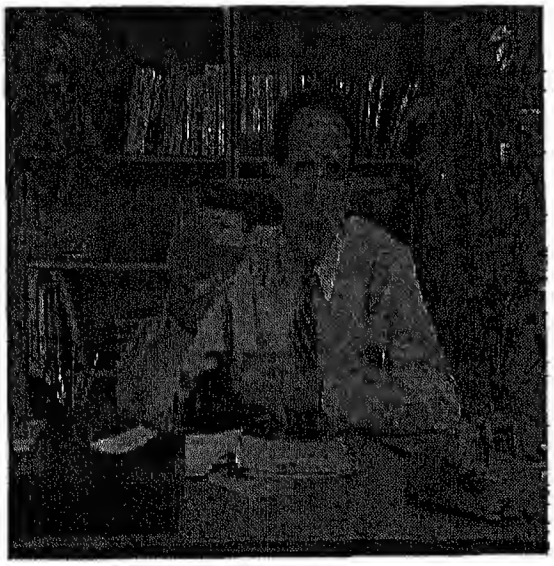


(a) Original

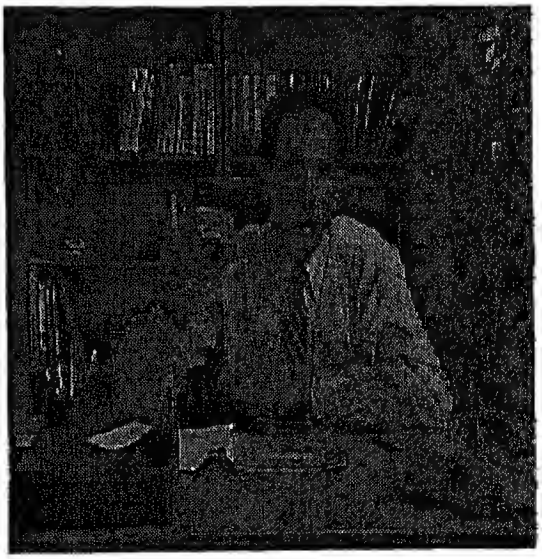


(b) Reconstructed

Figure 6 29 Original and Reconstructed *SALESMAN* Sequence Frame-9

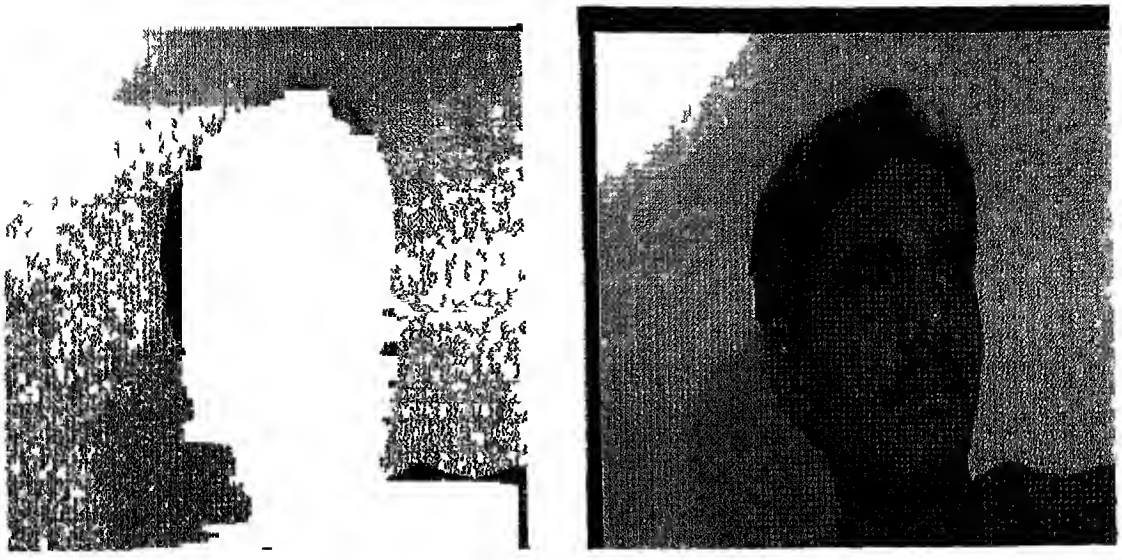


(a) Original



(b) Reconstructed

Figure 6 30 Original and Reconstructed *SALESMAN* Sequence Frame 10

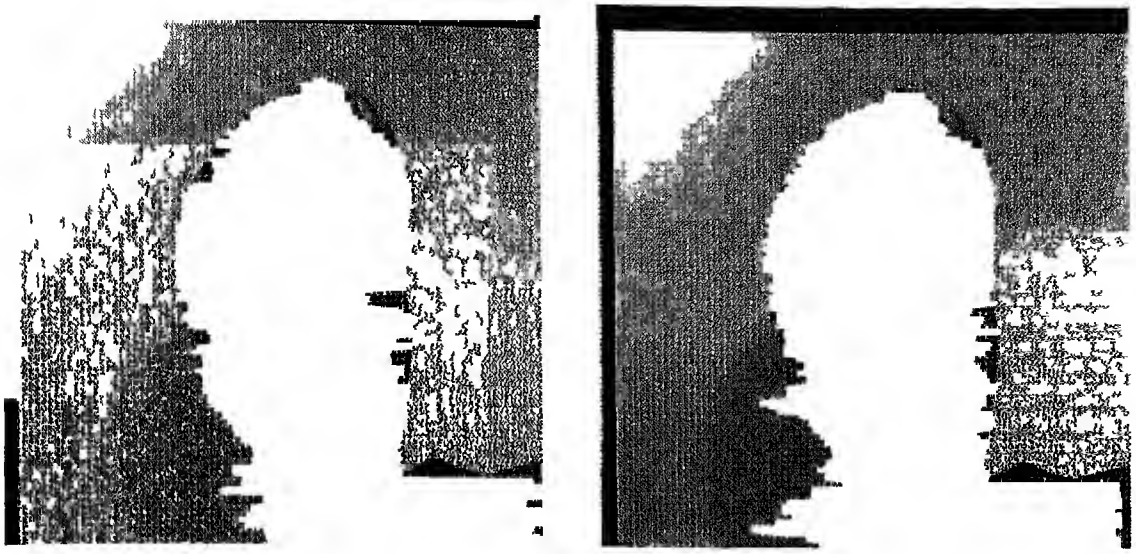


(a) Original (b) Reconstructed

Figure 6.31 Original and Reconstructed *AUTHOR* Sequence Frame-1

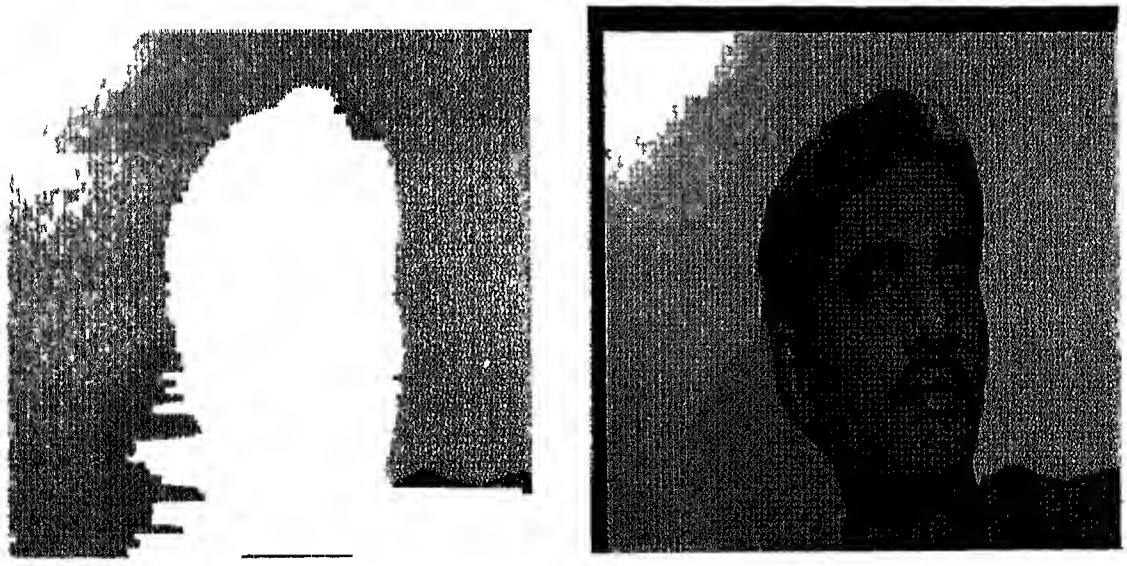
(a) Original (b) Reconstructed

Figure 6.32 Original and Reconstructed *AUTHOR* Sequence Frame-2



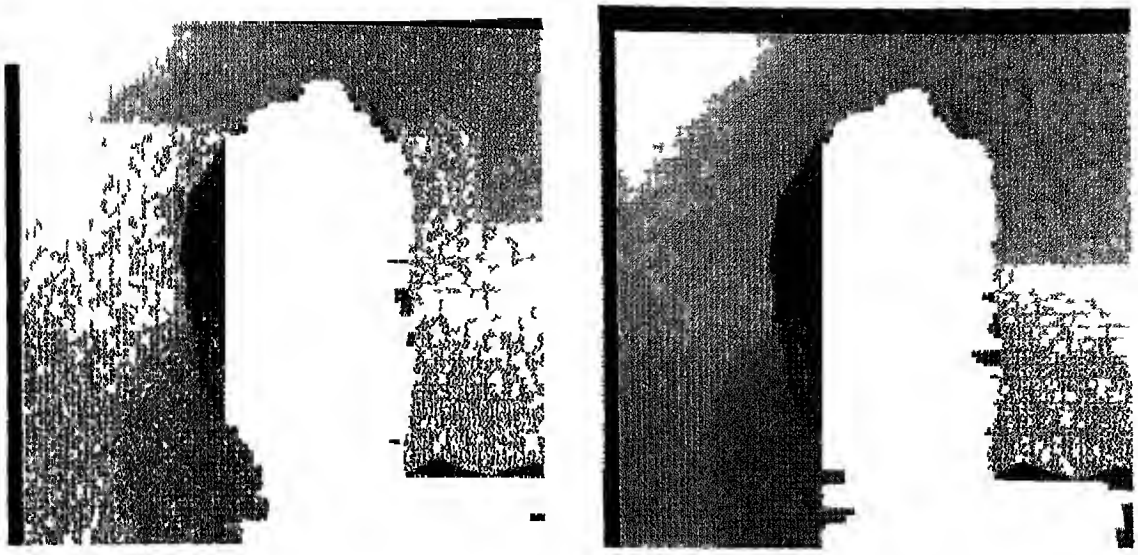
(a) Original (b) Reconstructed

Figure 6.33 Original and Reconstructed *AUTHOR* Sequence Frame 3



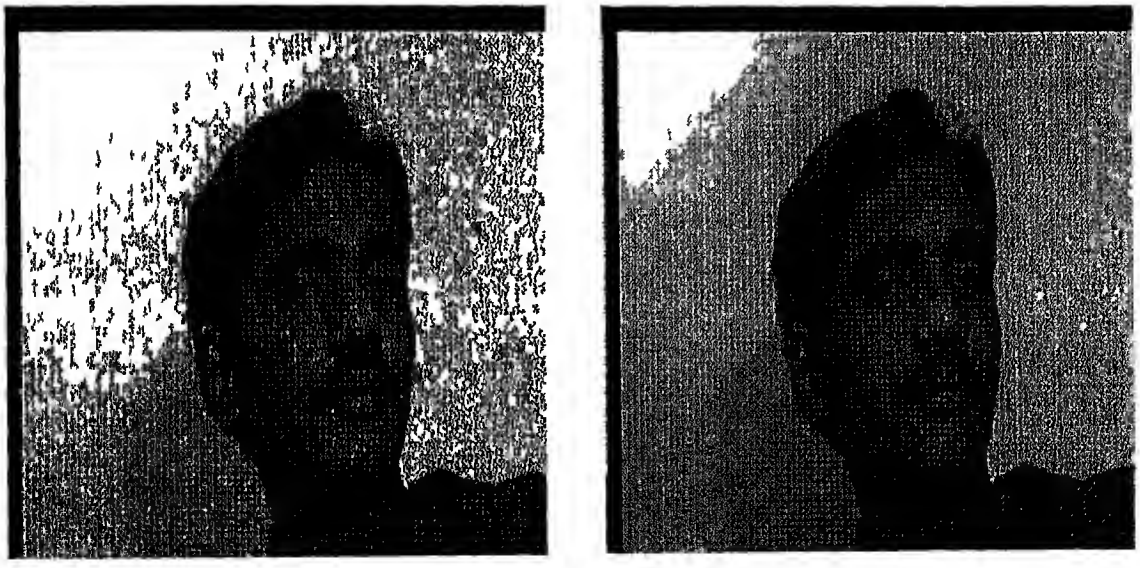
(a) Original (b) Reconstructed

Figure 6.34 Original and Reconstructed *AUTHOR* Sequence Frame-4



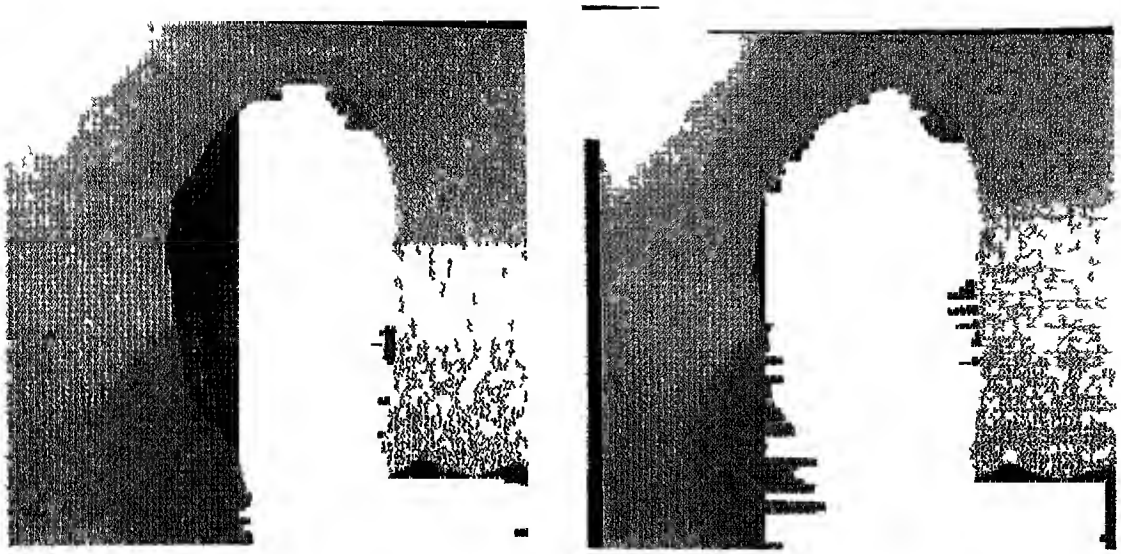
(a) Original (b) Reconstructed

Figure 6 35 Original and Reconstructed *AUTHOR* Sequence Frame 5



(a) Original (b) Reconstructed

Figure 6 36 Original and Reconstructed *AUTHOR* Sequence Frame-6

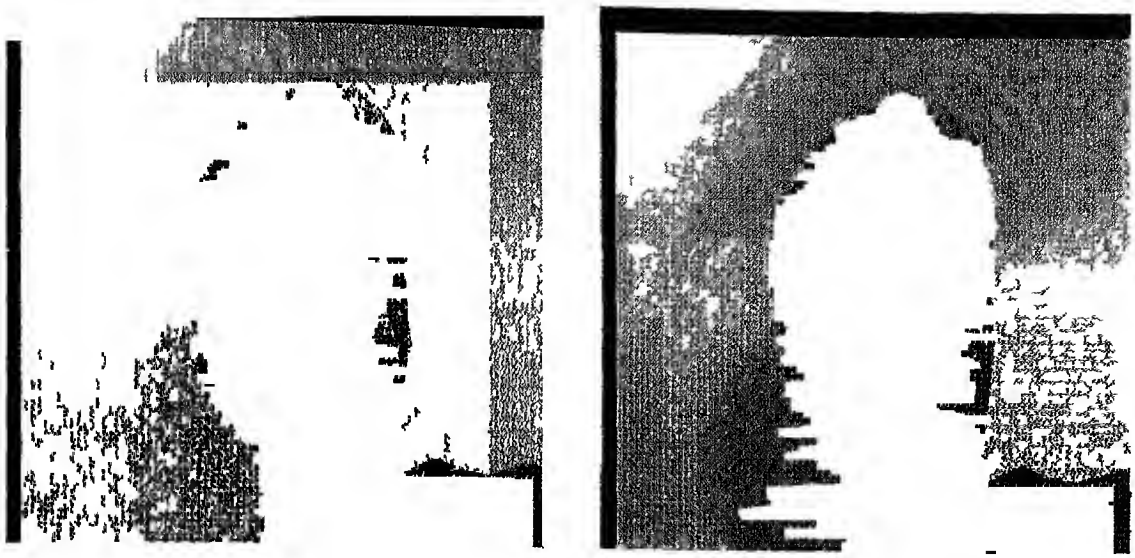


(a) Original (b) Reconstructed

Figure 6.37 Original and Reconstructed *AUTHOR* Sequence Frame 7

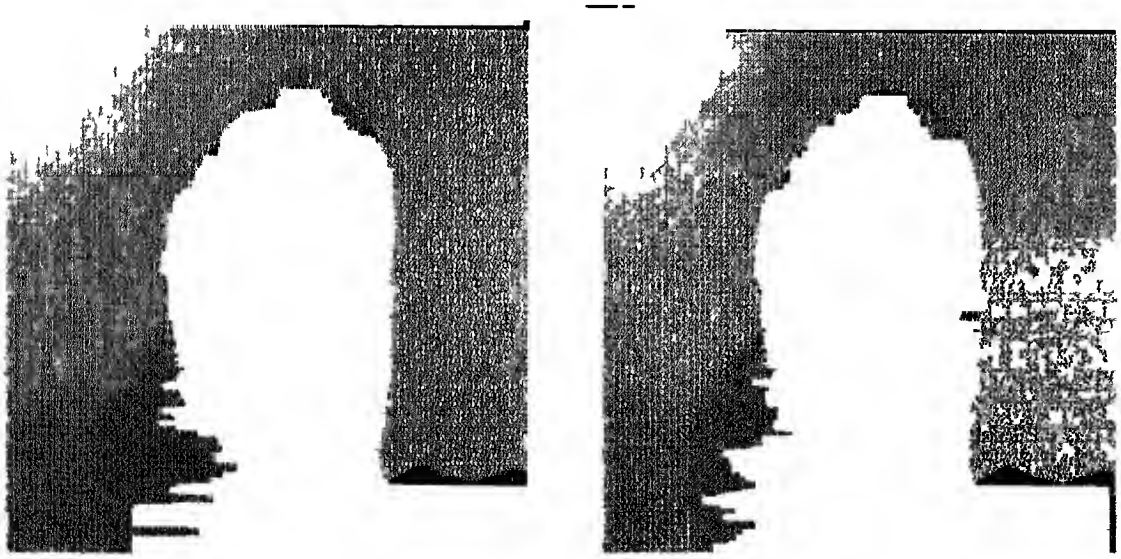
(a) Original (b) Reconstructed

Figure 6.38 Original and Reconstructed *AUTHOR* Sequence Frame 8



(a) Original (b) Reconstructed

Figure 6.39 Original and Reconstructed *AUTHOR* Sequence Frame 9



(a) Original (b) Reconstructed

Figure 6.40 Original and Reconstructed *AUTHOR* Sequence Frame 10

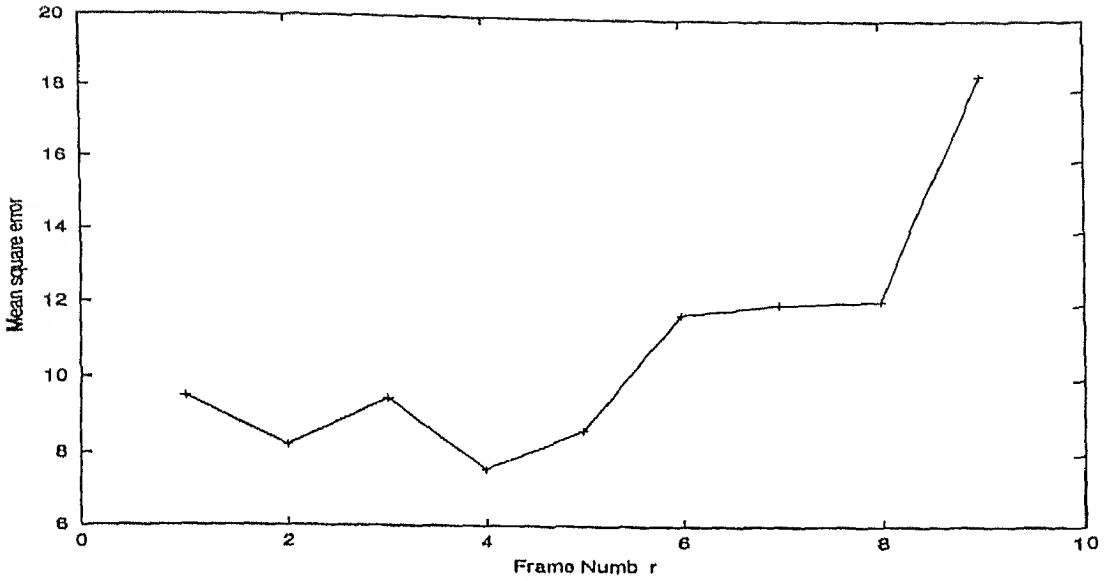


Figure 6.41 Mean square error of the reconstructed frames of *CLAIRE* Sequence

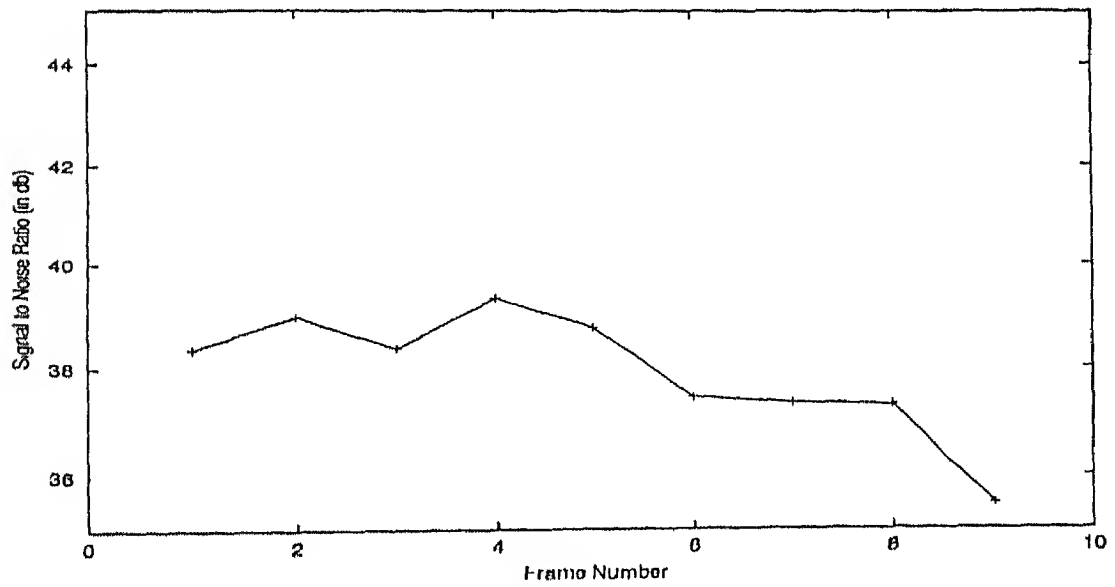


Figure 6.42 Signal to Noise Ratio of the reconstructed frames of *CLAIRE* Sequence

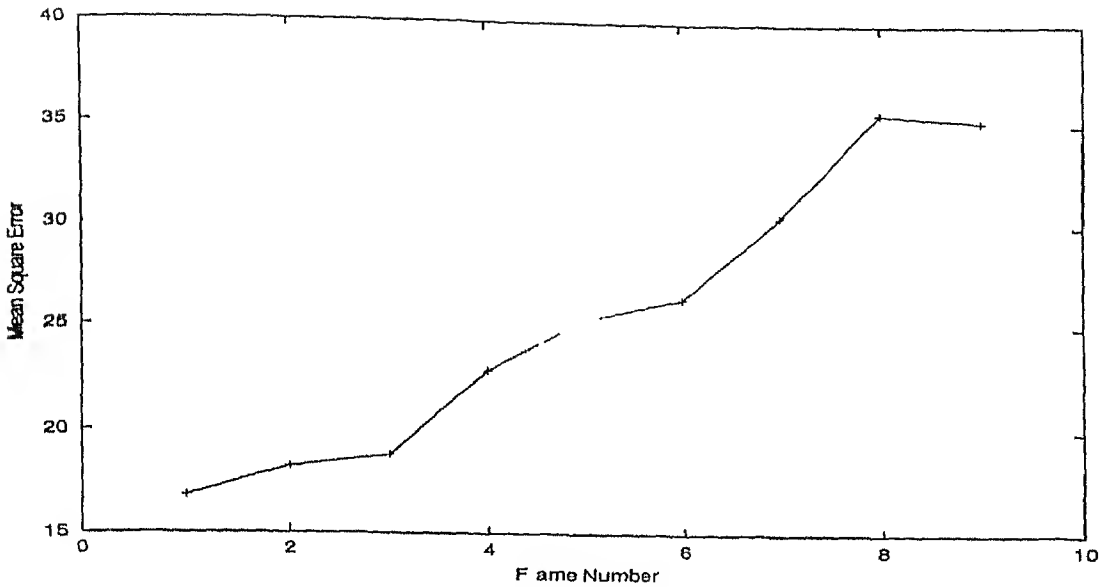


Figure 6 43 Mean square error of the reconstructed frames of *SALESMAN* Sequence

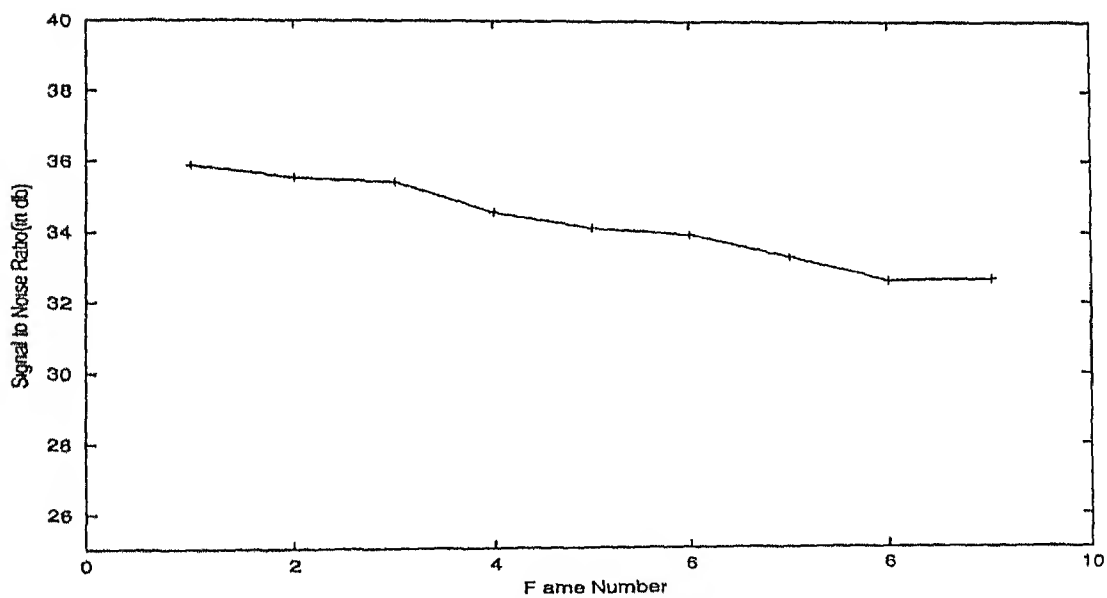


Figure 6 44 Signal to Noise ratio of the reconstructed frames of *SALESMAN* Sequence

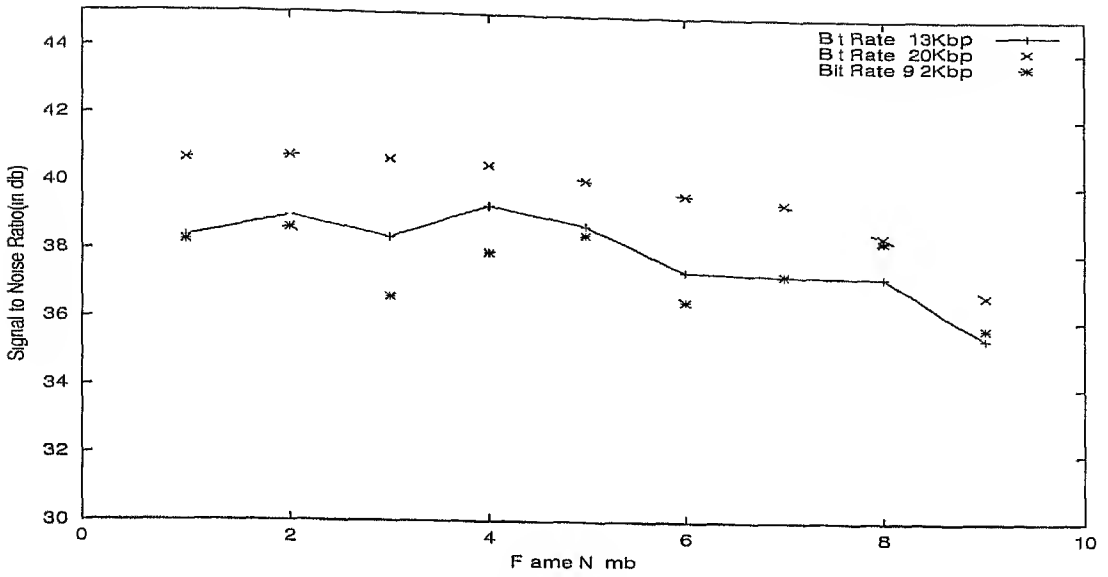


Figure 6.45 Comparision of SNR at Various Bit Rates for *CLAIRE* Sequence

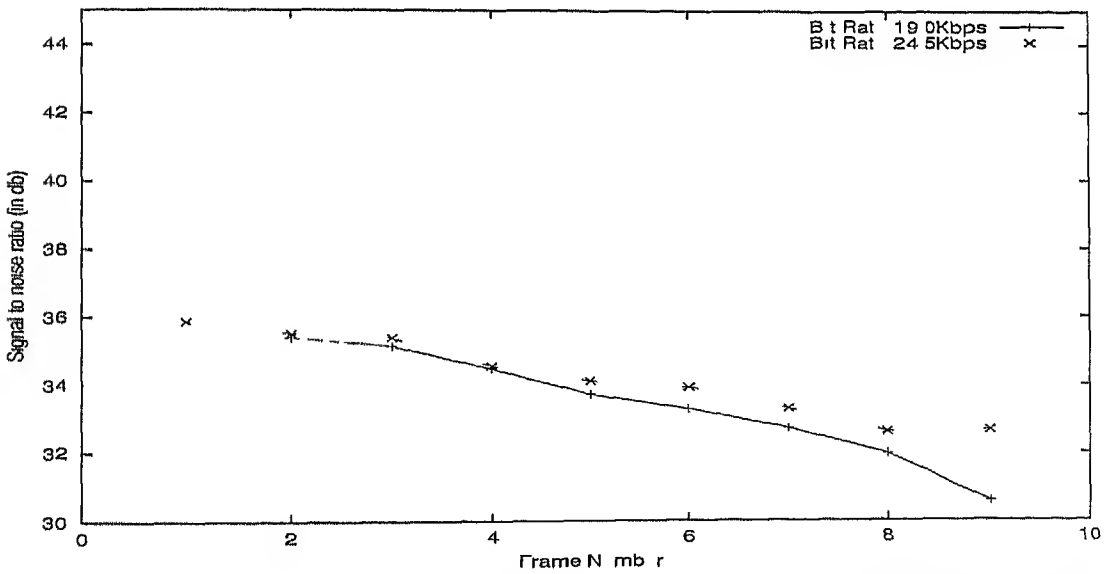


Figure 6.46 Comaprision of SNR at Various Bit Rates for *SALESMAN* Sequence

Chapter 7

CONCLUSIONS AND SCOPE FOR FUTURE WORK

7 1 Conclusions

In this thesis we described the implementation of a low bit rate video codec for a bit rate varying between 10 to 20 Kbps It is a region based technique employing morphological watershed algorithm for segmentation Unlike other segmentation methods, watershed algorithm provides accurate segmentation at low computational cost The oversegmentation arising from applying a simple gradient image as input to the watershed algorithm is taken care of by using multiscale gradient followed by an algorithm to eliminate small local minima In addition, the segmentation based method eliminates the problem of transmitting large amount of region shape information because this can be obtained by segmenting the reconstructed picture at both the ends

The motion estimation algorithm used in the design of the coder uses a novel

corner detection and tracking approach This motion estimation approach is much more efficient than other suggested approaches which use optimisation methods Corner points are used to represent a region and estimation of their motion, through tracking, is used to characterize the motion of the region This offers two major advantages lesser computational cost and lower noise susceptibility

The error image obtained by subtracting the motion compensated frame from the current frame is coded using the DCT The error information of only those significant error blocks is sent for which the error is greater than a threshold The algorithm developed was tested on first 10 frames of standard Claire Salesman sequence as well as the Author sequence A data rate between 10 to 20 Kbps was obtained at a frame rate of 7 5 frames per second The SNR was of the order of 31 38 dB The Low bit rate coder developed is robust and computationally efficient

7 2 Scope for future work

The segmentation process is controlled by the parameter h Fixing a value of h depending on one sequence may not give very good segmentation for other sequences One may find an optimum value for h through experimentation Fixing up small values like 8 or 10 for h may be suitable for some sequences in which case we call the segmentation algorithm as unsupervised, but will not give optimum segmentation results for all sequences This may lead to over segmentation As a result the moving regions will increase and the bit rate will go high So an automatic detection method for this constant h is needed in which case the segmentation algorithm will become fully unsupervised

In motion estimation, corner tracking must have sufficient immunity to

outliers If the number of outliers are more especially when the motion is large between the frames least square method may not converge In such cases outliers create serious problems and motion estimation may fail Also efficient method is needed to fill the Uncovered regions produced after motion compensation Extension of this approach to color sequences requires good segmention algorithm for color images [6]

Appendix-A
Algorithm Fast Watersheds

```
#define MASK -2 /* initial value of a threshold level */
#define WSHED 0 /* value of the pixels belonging to the watersheds*/
#define INIT -1

--INPUT    imi  decimal image
--OUTPUT   imo  image of the labeled watersheds

* INITIALIZATIONS
    -  Value INIT is assigned to each pixel of imo  imo(p) = INIT
    -  current_label = 0
    -  current_dist  integer variable
    -  imd   work image (of distances), initalized to 0
* Sort the pixels of imi, in the increasing order of
                                        their gray values
    Let h(min) and h(max) designate the lowest and highest
                            values respectively
* For h = h(min) to h(max) {
        /* geodesic SKIZ of level h-1 inside level h */
  For every pixel p such that imi(p) = h {
        /* These pixels are accessed directly through
                                    the sorted array  */
```

```
imo(p) = MASK
if there exists p' in N(G)(p) such that imo(p )
                              >  0 or imo(p') = WSHED
{
   imd(p) = 1
   fifo_add(p)
 }
 current_dist = 1  fifo_add(fictitious_pixel)
 repeat indefinitely {
   p = fifo_first()
   if p = fictitious_pixel{
        if fifo_empty() = true then BREAK
        else { fifo_add(fictitious_pixel)
        current_dist = current_dist + 1
        p = fifo_first(),
    }
  }
  For every pixel p' in N(G)(p) {
        if imd(p') < current_dist and (imo(p')
                     > 0 or imo(p') = WSHED){
            /* ie e   p' belongs to an already
              labeled basin or to the Watersheds */
           if imo(p') > 0 {
            if imo(p') = MASK or imo(p) = WSHED then
               imo(p) = imo(p')
            else if imo(p) != imo(p') then
                imo(p) = WSHED
```

```
                }
                else if imo(p) = MASK then imo(p) = WSHED
            }
        }
    }
    /* checks if new minima have been discovered */
    imd(p) = 0    /* the distance associated with p is reset to 0 */

    if (imd(p) = MASK {
        current_label = current_label +1
        fifo_add(p)    imo(p) = current_label
        while fifo_empty() = false {
            p' = fifo_first(),
            For every pixel p'' in N(G)(p') {
                if imo(p'') = MASK { fifo_add(p'')
                imo(p'') = current_label    }
            }
        }
    }
  }
  }
```

Appendix-B
VLC Tables for Coding the DCT Coefficients

Table B 1 Huffman code table for luminance DC difference in JPEG

category	code length	Codeword
0	2	00
1	3	010
2	3	011
3	3	100
4	3	101
5	3	110
6	4	1110
7	5	11110
8	6	111110
9	7	1111110
10	8	11111110
11	9	111111110

Table B 2 VLC Table for Luminance AC coefficients in JPEG

Run	Level	Codeword	Run	Level	Codeword
0	0	1010	3	5	1111111110010000
0	1	00	3	6	1111111110010001
0	2	01	3	7	1111111110010010
0	3	100	3	7	1111111110010010
0	4	1011	3	8	1111111110010011
0	5	11010	3	9	1111111110010100
0	6	1111000	4	1	111011
0	7	11111000	4	2	1111111000
0	8	1111110110	4	3	1111111110010110
0	9	1111111110000010	4	4	1111111110010111
0	10	1111111110000011	4	5	1111111110011000
1	1	1100	4	6	1111111110011001
1	2	11011	4	7	1111111110011010
1	3	1111001	4	8	1111111110011011
1	4	111110110	4	9	1111111110011100
1	5	11111110110	4	10	1111111110011101
1	6	1111111110000100	5	1	1111010
1	7	1111111110000101	5	2	11111110111
1	8	1111111110000110	5	3	1111111110011110
1	9	1111111110000111	5	4	1111111110011111

Run	Level	Codeword	Run	Level	Codeword
1	10	1111111110001000	5	5	1111111110010000
2	1	11100	5	6	1111111110010001
2	2	11111001	5	7	1111111110010010
2	3	1111110111	5	8	1111111110010011
2	4	111111110100	5	9	1111111110010100
2	5	1111111110001001	5	10	1111111110010101
2	6	1111111110001010	6	1	1111011
2	7	1111111110001011	6	2	111111110110
2	8	1111111110001100	6	3	1111111110100110
2	9	1111111110001101	6	4	1111111110100111
2	10	1111111110001110	6	5	1111111110101000
3	1	111010	6	6	1111111110101001
3	2	111110111	6	7	1111111110101010
3	3	111111110101	6	8	1111111110101011
3	4	1111111110001111	6	9	1111111110101100
6	10	1111111110101101	10	6	1111111111001011
7	1	11111010	10	7	1111111111001100
7	2	111111110111	10	8	1111111111001101
7	3	1111111110101110	10	9	1111111111001110
7	4	1111111110101111	10	10	1111111111001111
7	5	1111111110110000	11	1	1111111001
7	6	1111111110110001	11	2	1111111111010000
7	7	1111111110110010	11	3	1111111111010001
7	8	1111111110110011	11	4	1111111111010010
7	9	1111111110110100	11	5	1111111111010011
7	10	1111111110110101	11	6	1111111111010100
8	1	111111000	11	7	1111111111010101
8	2	111111111000000	11	8	1111111111010110
8	3	1111111110110110	11	9	1111111111010111
8	4	1111111110110111	11	10	1111111111011000

Run	Level	Codeword	Run	Level	Codeword
8	5	1111111110111000	12	1	1111111010
8	6	1111111110111001	12	2	1111111111011001
8	7	1111111110111010	12	3	1111111111011010
8	8	1111111110111011	12	4	1111111111011011
8	9	1111111110111100	12	5	1111111111011100
8	10	1111111110111101	12	6	1111111111011101
9	1	111111001	12	7	1111111111011110
9	2	1111111110111110	12	8	1111111111011111
9	3	1111111110111111	12	9	1111111111100000
9	4	1111111111000000	12	10	1111111111100001
9	5	1111111111000001	13	1	11111111000
9	6	1111111111000010	13	2	1111111111100010
9	7	1111111111000011	13	3	1111111111100011
9	8	1111111111000100	13	4	1111111111100100
9	9	1111111111000101	13	5	1111111111100101
9	10	1111111111000110	13	6	1111111111100110
10	1	111111010	13	7	1111111111100111
10	2	1111111111000111	13	8	1111111111101000
10	3	1111111111001000	13	9	1111111111101001
10	4	1111111111001001	13	10	1111111111101010
10	5	1111111111001010	14	1	1111111111101011
14	2	1111111111101100	15	1	1111111111110101
14	3	1111111111101101	15	2	1111111111110110
14	4	1111111111101110	15	3	1111111111110111
14	5	1111111111101111	15	4	1111111111111000
14	6	1111111111110000	15	5	1111111111111001
14	7	1111111111110001	15	6	1111111111111010
14	8	1111111111110010	15	7	1111111111111011
14	9	1111111111110011	15	8	1111111111111100
14	10	1111111111110100	15	9	1111111111111101
15	0	11111111001	15	10	1111111111111110

Bibliography

[1] Yutaka Yokoyama,Yoshihiro Miyamoto, and Mutsumi Ohta, "Very Low Bit Rate Video Coding Using Arbitrarily Shaped Region Based Motion Compensation," *IEEE Transactions On Circuits And Systems For Video Technology,* year Dec 1996, vol 5, no 6, pp 500 508

[2] Vincent L and Soille P, "Watersheds in digital spaces An efficient algorithm based on immersion simulations," *IEEE Trans Pattern Analysis Mach Intell* year 1991, vol 13, pp 583 598

[3] Demin Wang " A Multiscale Gradient Algorithm For Image Sementation Using Watersheds," *Pattern Recognition* year 1997, vol 30 no 12 pp 2043 2052

[4] Philippe Salembier and Montse Pardas, " Hierarchical Morphological Segmentation for Image Sequence Coding" *IEEE Trans on Image Processing,* year 1994, vol 3, no 5, pp 639 651

[5] Demin Wang "Unsupervised Video Segmentation and Object Tracking Using Modified Watershed Algorithm," *Workshop on Very Low Bit Video Coding "VLBV98", Beckman Institute fro Advanced Science and Technology, University of Illinois,* year 1998, pp 129 132

[6] Leila Shafarenko Maria Petrou Automatic Watershed Segmentation of Randomly Textured Color Images," *IEEE Transactions on Image Processing,* Nov 1997, vol 6 no 11 pp 1530 1544

[7] Larry S Shapiro " Affine Analysis of Image Sequences " *Ph D Thesis Sharp Laboratories of Europe, Oxford*

[8] Maiti N, Desai U B, Ray A K, Gantayet A M, "Mathematical Morphology An Image Processing Tool in Measurement of Droplet Size Distribution in Dropwise Condensation ' *Indian Conference on Computer Vision, Graphics and Image Processing* year December 1998, India, pp 446 451

[9] Stanley R Steinberg, Grayscale Morphology *Computer Vision Graphics and Image Processing,* year 1986, vol 35, pp 333 355

[10] Fu K S, Mui J K, ' A Survey On Image Segmentation," *Pattern Recognition,* year 1981, vol 13, pp 3 16

[11] Luc Vincent, " Morphological Grayscale Reconstruction in Image Analysis Applications and Efficient Algorithms," *IEEE Trans on Image Processing,* April 1993, vol 2, no 2, pp 176 201

[12] *Hand Book of Computer Vision,* Prentice Hall, New Jersey, year 1998

[13] Rao K R, Wang J J, *Techniques and Standards for Image, Video and Audio Coding,* Prentice Hall, New Jersey, year 1996

[14] Press W, *Numerical Recipes in C,* Cambridge University Press, year 1992

[15] Gonzalez and Richard F Woods, *Digital Image Processing,* Addision Wesley year 1993

[16] Jain A K, *Fundamentals of Digital Image Processing* Prentice Hall, New Jersey, year 1997

[17] Jae S LIm, *Two Dimensional Signal and Image Peocessing,* Prentice Hall New Jersey, year 1990

[18] Murat Tekalp, *Digital Video Processing* Prentice Hall, Signal Processing Se ries

A 130801